

明 細 書

発明の名称：

プロセス実行システム、データ構造、およびプロセス実行プログラム

技術分野

[0001] 本発明は、プロセス実行システム、データ構造、およびプロセス実行プログラムに関する。

背景技術

[0002] 近年、業務の生産性向上のために、デジタル技術を活用した業務の自動化が求められている。業務の自動化は主に業務システムを開発することによって行われる。一般的に、業務システムの開発は次のような方法によって進められる。すなわち、対象の業務をビジネスプロセスモデルとしてモデル化する。そして、システム化するのに必要な要件を当該ビジネスプロセスモデルに基づいて抽出して業務システムを開発する方法である。ビジネスプロセスモデルを記述する手法としては、例えば、BPMN (Business Process Modeling Notation) がある (特許文献 1、非特許文献 1、非特許文献 2 および非特許文献 3 参照)。

[0003] 上述した方法では、ビジネスプロセスモデルを作成する段階と、ビジネスプロセスモデルに基づいて業務システムを開発する段階の主に 2 つの段階が必要である。そのため、業務内容の変更に伴う業務システムの改修には多大な時間とコストを要することが多い。よって、上述した方法によって開発される業務システムは可塑性 (変形し易い性質) が乏しく、ビジネス環境の変化に柔軟に適応するのが難しいという課題がある。

[0004] 上記課題を解決する方法として、所定の仕様で記述したビジネスプロセスモデルを実行可能な実行環境を用意する方法が考えられる (非特許文献 4、非特許文献 5 および非特許文献 6 参照)。この方法によれば、業務内容を所定の仕様でビジネスプロセスモデルに表現することで当該業務に含まれるプロセスを実行環境に実行させることができる。そのため、ビジネスプロセス

モデルに基づいて業務システムを開発する段階をなくすることができる。すなわち、当該実行環境によって実行されるビジネスプロセスモデルを変更することで業務内容の変更に対応できるから可塑性が向上する。

[0005] 一方で、Digital Transformation（非特許文献7参照）が進む中で社会全体が複雑化し、ビジネス環境の変化の速度が速くなっている。そうした変化の速度が速いビジネス環境に適応するためには、自動化の対象となる業務の知識や経験を有している現場の業務担当者の多くが、上述した実行環境で実行可能なビジネスプロセスモデルの作成や修正等を行えることが望ましい。

[0006] しかしながら、上述した実行環境で実行可能なビジネスプロセスモデルは、実行環境ごとに独自の仕様で記述する必要があることが多い。また、当該実行環境の導入には多大なコストが必要であることが多く、導入できるのは一部の大企業であることから、インターネット等を使用して得られる情報も限られている。そのため、上述した実行環境で実行可能なビジネスプロセスモデルの作成や修正等に必要な学習コストが高く、現場の業務担当者が当該ビジネスプロセスモデルの作成や修正等を行うのは容易ではない。また、実行環境を変えた場合には、変更後の実行環境で実行可能なビジネスプロセスモデルを作り直す作業が必要になる。そうした作業を避けるために、他に優れた実行環境が出てきても容易に乗り換えることができず、かえって可塑性を損なう可能性もある。そうしたことから、上述した方法を実現するシステムは一般にはあまり普及していない。

[0007] 一方で、ビジネスプロセスの自動化に利用可能なRPA（Robotic Process Automation）のクラウドサービスやアプリが急速に普及している。こうしたRPAのクラウドサービスやアプリを使用することによって、現場の業務担当者がビジネスプロセスの一部のプロセスを容易に自動化できるようになってきている。これにより、業務内容が変わった場合でも、業務担当者が自らRPAのクラウドサービスやアプリを操作することでビジネスプロセスの実行内容を変更することができるようになって

きている。

先行技術文献

特許文献

[0008] 特許文献1：特開2012-203702号公報

非特許文献

[0009] 非特許文献1：Howard Smith, Peter Fingar, “Business Process Management: The Third Wave”, Meghan Kiffer Pr, 2002

非特許文献2：David W. Enstrom, “A Simplified Approach to IT Architecture with Bpmn: A Coherent Methodology for Modeling Every Level of the Enterprise”, iUniverse, 2016

非特許文献3：Bruce Silver, “BPMN Method and Style, Second Edition, with BPMN Implementer’s Guide”, Cody-Cassidy Press, 2017

非特許文献4：Arthur H. M. ter Hofstede, Wil M. P. van der Aalst, Michael Adams, Nick Russell (eds), “Modern Business Process Automation: YAWL and its Support Environment”, Springer, 2009

非特許文献5：Bernd Ruecker, “Practical Process Automation: Orchestration and Integration in Microservices and Cloud Native Architectures”, O’Re

i l l y M e d i a , 2 0 2 1

非特許文献6: Jakob Freund, Bernd Rucker,
“Real-Life BPMN (4th edition): Includes an introduction to DMN”, Camunda, 2019

非特許文献7: Stolterman E., Fors A. C. (2004) Information Technology and the Good Life. In: Kaplan B., Truex D. P., Wastell D., Wood-Harper A. T., DeGross J. I. (eds) Information Systems Research. IFIP International Federation for Information Processing, vol 143. Springer, Boston, MA.

発明の概要

発明が解決しようとする課題

[0010] しかしながら、業務を構成しているビジネスプロセスのうち、RPAによって自動化されているプロセスと、他のプロセスとの間の関係を把握するのは難しく、RPAによって実行したプロセスが他のプロセスに対して予期しない影響を与えるなど、業務を安定的に行う上で課題がある。RPAによって自動化されているプロセスの存在や内容を、当該プロセスの業務担当者以外の担当者が知らない場合もあり、RPAによって実行されているプロセスの監視やメンテナンスを組織全体で行うのが難しいという課題もある。こうした課題は、RPAによって自動化されているプロセスと他のプロセスとの間の関係が複雑になるほどより顕著になり、安定した業務運営に支障をきたしかねない。

[0011] 本発明は、上記課題を鑑みてなされたものであり、可塑性と安定性とを両立させることによって変化の速度が速いビジネス環境への適応度が高いプロ

セス実行システムを提供することを目的とする。

課題を解決するための手段

[0012] 本発明の上記目的は、下記的手段によって達成される。

[0013] (1) プロセスの内容を定義したプロセス定義情報を生成するために使用されるプロセス生成情報を有するプロセスマスタノードを複数含み、当該複数のプロセスマスタノードによって生成されるプロセスノード同士の間の関係を表現したプロセスネットワークモデルから、前記プロセス定義情報を有するプロセスノードを複数含むプロセスネットワークを生成するプロセスネットワーク生成部と、

前記プロセスネットワークに含まれるプロセスノードの状態と、所定のプロセス状態遷移規則と、に基づいて、特定のプロセスノードを実行可能にするプロセスネットワーク状態管理部と、

前記プロセスネットワークに含まれる実行可能な前記プロセスノードが有する前記プロセス定義情報によって定義されたプロセスを実行するプロセスネットワーク実行部と、を有し、

前記プロセスネットワーク生成部は、前記プロセスネットワークモデルの入力を受け付けるプロセスネットワークモデル入力部を備え、当該プロセスネットワークモデル入力部が入力を受け付けた前記プロセスネットワークモデルから前記プロセスネットワークを生成し、

前記プロセス定義情報は、当該プロセス定義情報によって定義されたプロセスを実行するのに必要なアクションに関する情報を含み、

前記プロセスネットワーク実行部は、要求に応じて所定のアクションを実行するアクション実行システムに対して、実行可能な前記プロセスノードの前記プロセス定義情報が含む前記アクションに関する情報に基づいて当該アクションの実行要求を送信することによって、当該プロセス定義情報によって定義されたプロセスを実行する、プロセス実行システム。

[0014] (2) 前記プロセスネットワークモデル入力部は、前記複数のプロセスマスタノードによって生成される前記プロセスノード同士の間の関係を階層的

に表現した階層型のプロセスネットワークモデルの入力を受け付け、

前記プロセスネットワーク生成部は、前記プロセスネットワークモデル入力部が入力を受け付けた前記階層型のプロセスネットワークモデルに対応する階層型のプロセスネットワークを生成する、上記（１）に記載のプロセス実行システム。

[0015] （３）前記プロセスネットワーク状態管理部は、特定のプロセスノードの状態を変更するプロセス状態変更要求を受け付けるプロセス状態変更要求受付部を備え、当該プロセス状態変更要求受付部が受け付けた前記プロセス状態変更要求に基づいて前記プロセスネットワークに含まれる当該特定のプロセスノードの状態を変更する、上記（１）または上記（２）に記載のプロセス実行システム。

[0016] （４）一の前記プロセスマスタノードが有する前記プロセス生成情報を、当該一の前記プロセスマスタノードに対応する一の前記プロセスノードに前記プロセス定義情報として転写する転写プロセスを実行する転写部をさらに有し、

前記プロセス生成情報は、前記転写プロセスを定義した一または複数の転写プロセス定義情報を含み、

前記転写プロセス定義情報は、プロセス実行システムが通信可能な環境が前記転写プロセスに与える影響をモデル化した転写因子モデルを定義した一または複数の転写因子モデル定義情報を含み、

前記転写部は、前記転写プロセスの実行要求を受け付ける転写要求受付部と、前記転写因子モデル定義情報によって定義された前記転写因子モデルを評価する転写因子モデル評価部と、前記転写プロセス定義情報によって定義された前記転写プロセスを実行する転写プロセス実行部と、を備え、

前記転写プロセス実行部は、前記転写因子モデル評価部が評価した一または複数の前記転写因子モデルの評価結果に基づいて前記転写プロセスを制御する、上記（１）から（３）のいずれか１つに記載のプロセス実行システム。

- [0017] (5) 前記プロセスネットワーク実行部は、前記プロセス定義情報によって定義されたプロセスを実行する際に、前記転写プロセスの実行要求を前記転写要求受付部に送信することによって、当該プロセス定義情報を生成するために使用される前記プロセス生成情報を前記転写部が転写することによって更新された前記プロセス定義情報に基づいて当該プロセスを実行する、上記(4)に記載のプロセス実行システム。
- [0018] (6) 前記転写因子モデル評価部は、一の前記転写因子モデルを評価する際に、他の前記転写因子モデルの評価結果を当該一の前記転写因子モデルの入力データとして使用する、上記(4)または(5)に記載のプロセス実行システム。
- [0019] (7) 前記プロセス定義情報は、当該プロセス定義情報を有するプロセスが活性であるか非活性であるかを表現する活性フラグを含み、
前記プロセス生成情報は、前記活性フラグを生成するために使用される活性フラグマスタを含み、
前記活性フラグマスタは、前記転写因子モデルの評価結果に応じて前記活性フラグが定まるように構成された前記転写プロセス定義情報を含み、
前記プロセスネットワーク状態管理部は、前記プロセスノードが有する前記プロセス定義情報に含まれる前記活性フラグが非活性であることを表現している場合には、当該プロセスノードを存在しないものとして扱う、上記(4)から(6)のいずれか1つに記載のプロセス実行システム。
- [0020] (8) 前記プロセスネットワーク生成部は、前記プロセスネットワークモデルを出力するプロセスネットワークモデル出力部を有し、
前記プロセスネットワークモデル出力部は、前記プロセスネットワークモデルをシリアル化されたデータを出力する、上記(1)から(7)のいずれか1つに記載のプロセス実行システム。
- [0021] (9) 前記プロセスネットワークモデル入力部は、前記プロセスネットワークモデルをシリアル化されたデータの入力を受け付け、
前記プロセスネットワーク生成部は、前記プロセスネットワークモデル入

力部が入力を受け付けた前記プロセスネットワークモデルをシリアルライズしたデータをデシリアルライズすることによって得られた前記プロセスネットワークモデルを用いて前記プロセスネットワークを生成する、上記（８）に記載のプロセス実行システム。

[0022] （１０）前記プロセスネットワークモデル出力部は、前記プロセスネットワークモデルをシリアルライズする際に、前記プロセスネットワークモデルと、当該プロセスネットワークモデルをシリアルライズしたデータとが一对一に対応する規則に基づいてシリアルライズする、上記（８）または（９）に記載のプロセス実行システム。

[0023] （１１）前記プロセスネットワークモデル出力部が前記プロセスネットワークモデルをシリアルライズしたデータを出力する際に使用するデータ形式は、キーと値の組を配列したデータ形式である、上記（８）から（１０）のいずれか１つに記載のプロセス実行システム。

[0024] （１２）前記プロセスネットワークモデル出力部が前記プロセスネットワークモデルをシリアルライズしたデータを出力する際に使用するデータ形式は、ＪＳＯＮである、上記（１１）に記載のプロセス実行システム。

[0025] （１３）前記プロセスネットワークモデルを表示するプロセスネットワークモデル表示部をさらに有する、上記（１）から（１２）のいずれか１つに記載のプロセス実行システム。

[0026] （１４）前記プロセスネットワークを表示するプロセスネットワーク表示部をさらに有する、上記（１）から（１３）のいずれか１つに記載のプロセス実行システム。

[0027] （１５）プロセス実行システムに用いられるデータ構造であって、
プロセスの内容を定義したプロセス定義情報を有するプロセスノードを複数含むプロセスネットワークを生成するために使用されるプロセスネットワークモデルを含み、

前記プロセスネットワークモデルは、前記プロセス定義情報を生成するために使用されるプロセス生成情報を有するプロセスマスタノードを複数含み

、当該複数のプロセスマスタノードによって生成されるプロセスノード同士の間の関係を表現し、

前記プロセス定義情報は、当該プロセス定義情報によって定義されたプロセスを実行するのに必要なアクションに関する情報を含む、データ構造。

[0028] (16) 前記プロセスネットワークモデルは、前記複数のプロセスマスタノードによって生成される前記プロセスノード同士の間の関係を階層的に表現した階層型のプロセスネットワークモデルであり、当該階層型のプロセスネットワークモデルに対応する階層型のプロセスネットワークを生成するために使用される、上記(15)に記載のデータ構造。

[0029] (17) 前記プロセス生成情報は、当該プロセス生成情報を、当該プロセス生成情報を有する前記プロセスマスタノードに対応する前記プロセスノードに前記プロセス定義情報として転写する転写プロセスを定義した一または複数の転写プロセス定義情報を含み、

前記転写プロセス定義情報は、前記プロセス実行システムが通信可能な環境が前記転写プロセスに与える影響をモデル化した転写因子モデルを定義した一または複数の転写因子モデル定義情報を含む、上記(15)または(16)に記載のデータ構造。

[0030] (18) 一の前記転写因子モデルが他の前記転写因子モデルの評価結果を入力データとして設定可能に表現される、上記(17)に記載のデータ構造。

[0031] (19) 前記プロセス定義情報は、当該プロセス定義情報を有するプロセスが活性であるか非活性であるかを表現する活性フラグを含み、

前記プロセス生成情報は、前記活性フラグを生成するために使用される活性フラグマスタを含み、

前記活性フラグマスタは、前記転写因子モデルの評価結果に応じて前記活性フラグが定まるように構成された前記転写プロセス定義情報を含む、上記(17)または(18)に記載のデータ構造。

[0032] (20) キーと値の組を配列したデータ形式によって表現される、上記(

15) から (19) のいずれか 1 つに記載のデータ構造。

[0033] (21) JSON のデータ形式によって表現される、上記 (15) から (20) のいずれか 1 つに記載のデータ構造。

[0034] (22) プロセスの内容を定義したプロセス定義情報を生成するために使用されるプロセス生成情報を有するプロセスマスタノードを複数含み、当該複数のプロセスマスタノードによって生成されるプロセスノード同士の間の関係を表現したプロセスネットワークモデルの入力を受け付けるステップと、

前記プロセスネットワークモデルの入力を受け付けるステップにおいて入力を受け付けた前記プロセスネットワークモデルから、前記プロセス定義情報を有する前記プロセスノードを複数含むプロセスネットワークを生成するステップと、

前記プロセスネットワークに含まれる前記プロセスノードの状態と、所定のプロセス状態遷移規則と、に基づいて前記プロセスノードの状態遷移処理を行うステップと、

前記プロセスネットワークに含まれる実行可能な前記プロセスノードの実行処理を行うステップと、

を含む処理を、コンピュータに実行させるためのプロセス実行プログラム。

[0035] (23) 前記プロセスネットワークモデルの入力を受け付けるステップにおいて、前記複数のプロセスマスタノードによって生成される前記プロセスノード同士の間の関係を階層的に表現した階層型のプロセスネットワークモデルの入力を受け付け、

前記プロセスネットワークを生成するステップにおいて、前記プロセスネットワークモデルの入力を受け付けるステップにおいて入力を受け付けた前記階層型のプロセスネットワークモデルに対応する階層型のプロセスネットワークを生成する、上記 (22) に記載のプロセス実行プログラム。

[0036] (24) 前記プロセスノードの状態遷移処理を行うステップは、特定のプ

プロセスノードの状態を変更するプロセス状態変更要求を受け付けるステップと、当該プロセス状態変更要求を受け付けるステップにおいて受け付けた前記プロセス状態変更要求に基づいて前記プロセスネットワークに含まれる当該特定のプロセスノードの状態を変更する、上記（２２）または（２３）に記載のプロセス実行プログラム。

[0037] （２５）前記プロセス生成情報は、当該プロセス生成情報を、当該プロセス生成情報を有する前記プロセスマスタノードに対応する前記プロセスノードに前記プロセス定義情報として転写する転写プロセスを定義した一または複数の転写プロセス定義情報を含み、

前記転写プロセス定義情報は、プロセス実行システムが通信可能な環境が前記転写プロセスに与える影響をモデル化した転写因子モデルを定義した一または複数の転写因子モデル定義情報を含み、

前記転写プロセスの実行要求を受け付けるステップと、

前記転写因子モデル定義情報によって定義された前記転写因子モデルを評価するステップと、

前記転写プロセス定義情報によって定義された前記転写プロセスを実行するステップと、をさらに有し、

前記転写プロセスを実行するステップは、前記転写因子モデルを評価するステップにおける一または複数の前記転写因子モデルの評価結果に基づいて前記転写プロセスを制御する、上記（２２）から（２４）のいずれか１つに記載のプロセス実行プログラム。

[0038] （２６）前記プロセスノードの実行処理を行うステップは、前記転写プロセスの実行要求を送信することによって、当該プロセス定義情報を生成するために使用される前記プロセス生成情報を転写することによって更新された前記プロセス定義情報に基づいて当該プロセスを実行する、上記（２５）に記載のプロセス実行プログラム。

[0039] （２７）前記転写因子モデルを評価するステップにおいて、一の前記転写因子モデルを評価する際に、他の前記転写因子モデルの評価結果を当該一の

前記転写因子モデルの入力データとして使用する、上記（２５）または（２６）に記載のプロセス実行プログラム。

[0040] （２８）前記プロセス定義情報は、当該プロセス定義情報を有するプロセスが活性であるか非活性であるかを表現する活性フラグを含み、

前記プロセス生成情報は、前記活性フラグを生成するために使用される活性フラグマスタを含み、

前記活性フラグマスタは、前記転写因子モデルの評価結果に応じて前記活性フラグが定まるように構成された前記転写プロセス定義情報を含み、

前記プロセスノードの状態遷移処理を行うステップにおいて、前記プロセスノードが有する前記プロセス定義情報に含まれる前記活性フラグが非活性であることを表現している場合には、当該プロセスノードを存在しないものとして扱う、上記（２５）から（２７）のいずれか１つに記載のプロセス実行プログラム。

[0041] （２９）前記プロセスネットワークモデルを出力するステップをさらに有し、

当該プロセスネットワークモデルを出力するステップにおいて、前記プロセスネットワークモデルをシリアル化されたデータを出力する、上記（２２）から（２８）のいずれか１つに記載のプロセス実行プログラム。

[0042] （３０）前記プロセスネットワークモデルの入力を受け付けるステップにおいて、前記プロセスネットワークモデルをシリアル化されたデータの入力を受け付け、

前記プロセスネットワークを生成するステップにおいて、前記プロセスネットワークモデルの入力を受け付けるステップにおいて入力を受け付けた前記プロセスネットワークモデルをシリアル化されたデータをデシリアル化することによって得られた前記プロセスネットワークモデルを用いて前記プロセスネットワークを生成する、上記（２２）から（２９）のいずれか１つに記載のプロセス実行プログラム。

[0043] （３１）前記プロセスネットワークモデルを出力するステップにおいて、

前記プロセスネットワークモデルをシリアルライズする際に、前記プロセスネットワークモデルと、当該プロセスネットワークモデルをシリアルライズしたデータとが一对一に対応する規則に基づいてシリアルライズする、上記（２２）から（３０）のいずれか１つに記載のプロセス実行プログラム。

[0044] （３２）前記プロセスネットワークモデルを出力するステップにおいて、前記プロセスネットワークモデルをシリアルライズしたデータを、キーと値の組を配列したデータ形式によって出力する、上記（２２）から（３１）のいずれか１つに記載のプロセス実行プログラム。

[0045] （３３）前記プロセスネットワークモデルを出力するステップにおいて、前記プロセスネットワークモデルをシリアルライズしたデータを、ＪＳＯＮのデータ形式によって出力する、上記（３２）に記載のプロセス実行プログラム。

[0046] （３４）前記プロセスネットワークモデルを表示するステップをさらに有する、上記（２２）から（３３）のいずれか１つに記載のプロセス実行プログラム。

[0047] （３５）前記プロセスネットワークを表示するステップをさらに有する、上記（２２）から（３４）のいずれか１つに記載のプロセス実行プログラム。

発明の効果

[0048] 本発明に係るプロセス実行システムによれば、プロセスの内容を定義したプロセス定義情報を生成するために使用されるプロセス生成情報を有するプロセスマスタノードを複数含み、当該複数のプロセスマスタノードによって生成されるプロセスノード同士の間の関係を表現したプロセスネットワークモデルから、前記プロセス定義情報を有するプロセスノードを複数含むプロセスネットワークを生成するプロセスネットワーク生成部と、前記プロセスネットワークに含まれるプロセスノードの状態と、所定のプロセス状態遷移規則と、に基づいて、特定のプロセスノードを実行可能にするプロセスネットワーク状態管理部と、前記プロセスネットワークに含まれる実行可能な前

記プロセスノードが有する前記プロセス定義情報によって定義されたプロセスを実行するプロセスネットワーク実行部と、を有し、前記プロセスネットワーク生成部は、前記プロセスネットワークモデルの入力を受け付けるプロセスネットワークモデル入力部を備え、当該プロセスネットワークモデル入力部が入力を受け付けた前記プロセスネットワークモデルから前記プロセスネットワークを生成し、前記プロセス定義情報は、当該プロセス定義情報によって定義されたプロセスを実行するのに必要なアクションに関する情報を含み、前記プロセスネットワーク実行部は、要求に応じて所定のアクションを実行するアクション実行システムに対して、実行可能な前記プロセスノードの前記プロセス定義情報が含む前記アクションに関する情報に基づいて当該アクションの実行要求を送信することによって、当該プロセス定義情報によって定義されたプロセスを実行する。これにより、可塑性と安定性とを両立させることによって変化の速度が速いビジネス環境への適応度が高いプロセス実行システムを提供できる。

図面の簡単な説明

[0049] [図1]本発明の第1実施形態に係るプロセス実行システムと、当該プロセス実行システムと通信するアクション実行システムの構成を示す図である。

[図2]図1に示したプロセス実行システムのブロック図である。

[図3]図1に示したプロセス実行システムの制御部の機能ブロック図である。

[図4]図1に示したプロセス実行システムのプロセスネットワーク生成部の機能ブロック図である。

[図5]図1に示したプロセス実行システムのプロセスネットワークモデル入力部が入力を受け付けるプロセスネットワークモデルのデータ構造を例示する模式図である。

[図6]図1に示したプロセス実行システムのプロセスネットワーク生成部によって生成されるプロセスネットワークのデータ構造を例示する模式図である。

[図7]図1に示したプロセス実行システムのプロセスネットワークモデル入力

部が入力を受け付けるプロセスネットワークモデルのプロセスマスタノードのデータ構造を例示する模式図である。

[図8]図1に示したプロセス実行システムのプロセスネットワーク生成部によって生成されるプロセスネットワークのプロセスノードのデータ構造を例示する模式図である。

[図9]図1に示したプロセス実行システムのプロセスネットワークモデル入力部が入力を受け付けるプロセスネットワークモデルのプロセスマスタノードのアクション情報マスタのデータ構造を例示する模式図である。

[図10]図1に示したプロセス実行システムのプロセスネットワーク生成部によって生成されるプロセスネットワークのプロセスノードのアクション情報のデータ構造を例示する模式図である。

[図11]図1に示したプロセス実行システムにおいて実行される処理の手順を表すフローチャートである。

[図12]図1に示したプロセス実行システムにおいて実行されるプロセスノードの状態遷移処理の手順を表すフローチャートである。

[図13]図1に示したプロセス実行システムにおいて実行される親プロセスノードの配下のプロセスノード群に対する状態遷移処理の手順を表すフローチャートである。

[図14]図1に示したプロセス実行システムにおいて実行される最小のプロセス番号を有するプロセスノード群に対する状態遷移処理の手順を表すフローチャートである。

[図15]図1に示したプロセス実行システムにおいて実行される実行可能なプロセスノードの実行処理の手順を表すフローチャートである。

[図16]図1に示したプロセス実行システムにおいて実行されるプロセスノードの状態遷移処理を行うステップの動作例について説明するために使用するプロセスネットワークのデータ構造を示す模式図である。

[図17]図1に示したプロセス実行システムにおいて実行されるプロセスノードの状態遷移処理を行うステップの動作例について説明するために使用する

プロセスノードの状態遷移を表す図である。

[図18]本発明の第2実施形態に係るプロセス実行システムと、当該プロセス実行システムと通信するアクション実行システムの構成を示す図である。

[図19]図18に示したプロセス実行システムのブロック図である。

[図20]図18に示したプロセス実行システムの制御部の機能ブロック図である。

[図21]図18に示したプロセス実行システムのプロセスネットワーク状態管理部の機能ブロック図である。

[図22]図18に示したプロセス実行システムのプロセスネットワークモデル入力部が入力を受け付けるプロセスネットワークモデルのデータ構造を例示する模式図である。

[図23]図18に示したプロセス実行システムのプロセスネットワーク生成部によって生成されるプロセスネットワークのデータ構造を例示する模式図である。

[図24]図18に示したプロセス実行システムのプロセスネットワークモデル入力部が入力を受け付けるプロセスネットワークモデルのプロセスマスタノードのデータ構造を例示する模式図である。

[図25]図18に示したプロセス実行システムのプロセスネットワーク生成部によって生成されるプロセスネットワークのプロセスノードのデータ構造を例示する模式図である。

[図26]図18に示したプロセス実行システムにおいて実行される処理の手順を表すフローチャートである。

[図27]図18に示したプロセス実行システムにおいて実行される実行可能なプロセスノードの実行処理の手順を表すフローチャートである。

[図28]本発明の第3実施形態に係るプロセス実行システムの制御部の機能ブロック図である。

[図29]本発明の第3実施形態に係るプロセス実行システムの転写部の機能ブロック図である。

[図30]本発明の第3実施形態に係るプロセス実行システムのプロセス生成情報を示す概念図である。

[図31]本発明の第3実施形態に係るプロセス実行システムと環境との間の相互作用を示す概念図である。

[図32]本発明の第3実施形態に係るプロセス実行システムが参照するケースデータテーブルのデータ構造を例示する模式図である。

[図33]本発明の第3実施形態に係るプロセス実行システムのプロセスネットワークモデル入力部が保存するコンテキストデータのデータ構造を例示する模式図である。

[図34]本発明の第3実施形態に係るプロセス実行システムの転写プロセスタグの構文を例示する図である。

[図35]本発明の第3実施形態に係るプロセス実行システムの選択プロセスタグの構文を例示する図である。

[図36]本発明の第3実施形態に係るプロセス実行システムの反復プロセスタグの構文を例示する図である。

[図37]本発明の第3実施形態に係るプロセス実行システムの置換プロセスタグの構文を例示する図である。

[図38]本発明の第3実施形態に係るプロセス実行システムのプロセス因子モデルタグの構文を例示する図である。

[図39]本発明の第3実施形態に係るプロセス実行システムのプロセス因子モデルの出力データを例示する図である。

[図40]本発明の第3実施形態に係るプロセス実行システムにおいて実行される処理の手順を表すフローチャートである。

[図41]本発明の第3実施形態に係るプロセス実行システムにおいて実行される実行可能なプロセスノードの実行処理の手順を表すフローチャートである。

[図42]本発明の第3実施形態に係るプロセス実行システムのプロセス因子モデルにおいて実行される処理の手順を表すフローチャートである。

[図43]本発明の第3実施形態に係るプロセス実行システムの転写部において実行される評価対象転写因子モデルの評価処理の手順を表すフローチャートである。

[図44]本発明の第3実施形態に係るプロセス実行システムの転写部において実行される原評価対象転写因子モデルの評価処理の手順を表すフローチャートである。

[図45]本発明の第3実施形態に係るプロセス実行システムの転写部の動作例について説明するために使用するプロセス生成情報を例示する模式図である。

[図46]本発明の第3実施形態に係るプロセス実行システムの転写部の動作例について説明するために使用する転写情報を例示する模式図である。

[図47]本発明の第3実施形態に係るプロセス実行システムの転写部の動作例について説明するために使用する転写情報を例示する模式図である。

[図48]本発明の第3実施形態に係るプロセス実行システムの転写部における評価対象転写因子モデルの評価処理の動作例について説明するために使用する転写プロセスタグを例示する模式図である。

[図49]本発明の第3実施形態に係るプロセス実行システムの転写部における評価対象転写因子モデルの評価処理の動作例について説明するために使用する転写プロセスタグを例示する模式図である。

[図50]本発明の第3実施形態に係るプロセス実行システムの転写部における評価対象転写因子モデルの評価処理の動作例について説明するために使用する転写プロセスタグを例示する模式図である。

[図51]本発明の第3実施形態に係るプロセス実行システムの転写部における評価対象転写因子モデルの評価処理の動作例について説明するために使用する転写プロセスタグを例示する模式図である。

[図52]本発明の第3実施形態に係るプロセス実行システムの転写部における評価対象転写因子モデルの評価処理の動作例について説明するために使用する転写プロセスタグを例示する模式図である。

[図53]本発明の第3実施形態に係るプロセス実行システムの転写部における評価対象転写因子モデルの評価処理の動作例について説明するために使用する転写プロセスタグを例示する模式図である。

[図54]本発明の第3実施形態に係るプロセス実行システムの転写部の動作例について説明するために使用するアクション情報マスタを例示する模式図である。

[図55]本発明の第3実施形態に係るプロセス実行システムの転写部の動作例について説明するために使用するアクション情報を例示する模式図である。

[図56]本発明の第3実施形態に係るプロセス実行システムの転写部の動作例について説明するために使用するアクション情報マスタを例示する模式図である。

[図57]本発明の第3実施形態に係るプロセス実行システムの転写部の動作例について説明するために使用するアクション情報を例示する模式図である。

[図58]本発明の第3実施形態の改変例に係るプロセス実行システムのプロセスネットワーク生成部によって生成されるプロセスネットワークのデータ構造を例示する模式図である。

[図59]本発明の第3実施形態の改変例に係るプロセス実行システムのプロセスネットワークモデル入力部が入力を受け付けるプロセスネットワークモデルのデータ構造を例示する模式図である。

[図60]本発明の第3実施形態の改変例に係るプロセス実行システムのプロセスネットワークモデル入力部が入力を受け付けるプロセスネットワークモデルのプロセスマスタノードのデータ構造を例示する模式図である。

[図61]本発明の第3実施形態の改変例に係るプロセス実行システムのプロセスネットワーク生成部によって生成されるプロセスネットワークのプロセスノードのデータ構造を例示する模式図である。

[図62]本発明の第4実施形態に係るプロセス実行システムのプロセスネットワーク生成部の機能ブロック図である。

[図63]本発明の第4実施形態に係るプロセス実行システムのプロセスネット

ワークモデル出力部がプロセスネットワークモデルをシリアルライズしたデータを例示する模式図である。

[図64]本発明の第5実施形態に係るプロセス実行システムの制御部の機能ブロック図である。

[図65]本発明の第5実施形態に係るプロセス実行システムのプロセスネットワークモデル表示部の機能ブロック図である。

[図66]本発明の第5実施形態に係るプロセス実行システムのプロセスネットワーク表示部の機能ブロック図である。

[図67]本発明の第5実施形態に係るプロセス実行システムのプロセスネットワークモデル表示部がプロセスネットワークモデルを表示した様子を示す図である。

[図68]本発明の第5実施形態に係るプロセス実行システムのプロセスネットワーク表示部がプロセスネットワークを表示した様子を示す図である。

[図69]実施例に係るプロセス実行システムにおいて実行される処理について説明するために使用するプロセスネットワークモデルおよびプロセスネットワークの一部のデータ構造を示す模式図である。

[図70]実施例に係るプロセス実行システムにおいて実行される処理について説明するために使用するプロセスネットワークモデルおよびプロセスネットワークの一部のデータ構造を示す模式図である。

[図71]実施例に係るプロセス実行システムにおいて実行される処理について説明するために使用するプロセスネットワークモデルおよびプロセスネットワークの一部のデータ構造を示す模式図である。

[図72]実施例に係るプロセス実行システムにおいて実行される処理について説明するために使用するプロセスネットワークモデルおよびプロセスネットワークの一部のデータ構造を示す模式図である。

[図73]実施例に係るプロセス実行システムにおいて実行される処理について説明するために使用するプロセスネットワークモデルおよびプロセスネットワークの一部のデータ構造を示す模式図である。

[図74]実施例に係るプロセス実行システムにおいて実行される処理の一部を表すシーケンス図である。

[図75]実施例に係るプロセス実行システムにおいて実行される処理の一部を表すシーケンス図である。

[図76]実施例に係るプロセス実行システムにおいて実行される処理の一部を表すシーケンス図である。

[図77]実施例に係るプロセス実行システムにおいて実行される処理の一部を表すシーケンス図である。

[図78]実施例に係るプロセス実行システムにおいて実行される処理の一部を表すシーケンス図である。

発明を実施するための形態

[0050] 本発明の実施形態の説明に入る前に、本発明の技術的意義について概説する。近年、Digital Transformation（非特許文献7参照）が進む中で社会全体が複雑化し変化の速度が速くなっている（Bruno Latour, “Reassembling the Social: An Introduction to Actor-network-theory”, Oxford Univ Pr, 2007）。これに伴い、企業を取り巻くビジネス環境の変化の速度も速くなっている。企業活動における競争力を維持・向上させるためには、こうした変化の速度が速いビジネス環境への適応度を高める必要がある。そうした環境にあって、デジタル技術が急速に進歩していることから、デジタル技術の活用が企業の競争力を大きく左右するに至っている。特に、デジタル技術を活用した業務の自動化が急務になっており、業務の自動化を担う業務システムに対して、変化の速度が速いビジネス環境への適応度の向上が求められている。本発明の実施形態に係るプロセス実行システムが解決しようとする課題は、変化の速度が速いビジネス環境への適応度が高いプロセス実行システムを提供することにある。

[0051] 上述した課題を解決するプロセス実行システムは、業務内容の変化を容易

に吸収できる程度に変形し易い一方で、安定した業務運営をもたらすものでなければならない。すなわち、上述した課題を解決するプロセス実行システムは、可塑性と安定性という一見すると矛盾する関係をうまく調整して両立させる必要がある。本発明の実施形態に係るプロセス実行システムでは、可塑性と安定性とを両立させるために、プロセス同士の相互作用を表現するネットワークに着目する。具体的には、本発明の実施形態に係るプロセス実行システムは、各々のプロセスの実行に必要なアクションの実行をアクション実行システムに委譲し、実行可能なプロセスネットワーク (Executable Process Network)、特に実行可能なビジネスプロセスネットワーク (Executable Business Process Network) を用いた新しいビジネスプロセスモデルの実行環境を提供する。

[0052] プロセス同士の相互作用を表現するネットワークに着目する意義についてさらに詳しく説明する。Digital Transformation (非特許文献7参照)が進む中で人や機械などのあらゆるものが複雑なネットワークを介して相互作用する社会環境は複雑系として捉えることができる。複雑系をなすような環境、特に、構成要素間の相互作用に係る複雑性が非常に高く、予測困難な変動が起きやすく、(熱力学的な平衡状態に近い系と比較して)変化の速度も速いような環境に適応してきたシステムの例として、生物、特に多細胞生物のシステムがある (Stuart Kauffman, “At Home in the Universe: The Search for Laws of Self-Organization and Complexity”, Oxford Univ Pr, 1996; Stuart Kauffman, “A World Beyond Physics: The Emergence and Evolution of Life”, Oxford Univ Pr, 2019)。多細胞生物は、DNAに符号化された遺伝子の発現を調節することによって多様な細胞を協働させている (Bruce Alberts,

et al., “Molecular Biology of the Cell”, 6th Edition, Garland Science, 2014)。遺伝子の発現の調節には転写調節因子が関わっており、転写調節因子は複雑なネットワーク(以下、Transcription Network)を介して相互作用している(Uri Alon, “An Introduction to Systems Biology: Design Principles of Biological Circuits”, 2nd edition, Chapman and Hall/CRC, 2019)。多細胞生物を構成するタンパク質の種類は異なる生物種の間で共通のものが多い一方で、Transcription Networkは生物種の間で異なっている。多細胞生物のシステムは、モジュール(タンパク質)は共通化しつつ、モジュールを相互作用させるTranscription Networkなどを柔軟に変化させることで、外部環境Eoutの変化に応じてモジュールの組み合わせ方を柔軟に組み替えられるアーキテクチャを有していると考えられる。これにより、多細胞生物は、複雑系をなすような環境にうまく適応してきたと考えられる。

[0053] 本発明の実施形態に係るプロセス実行システムについて、多細胞生物との類推で述べれば、プロセスの内容を定義したプロセス定義情報はタンパク質(モジュール)に相当し、そのプロセス定義情報を生成するために使用されるプロセス生成情報は遺伝子に相当し、プロセス同士の間を表現したプロセスネットワークはTranscription Networkに相当し、転写因子モデルは転写調節因子などに相当する。本発明の実施形態に係るプロセス実行システムによれば、モジュールを共通化しつつ、モジュールを相互作用させるプロセスネットワークを調節することで、複雑系をなすような環境に適応し易くなる。

[0054] また、複雑系をなすような環境に適応するシステムの普遍的特徴として階層性が指摘されている(Herbert A. Simon, “The Sciences of the Artificial”, MIT

Press, 1996)。特に時間スケールの異なる階層間の相互作用の重要性は、古くはベルクソン哲学において指摘されており (Henri Bergson, “物質と記憶”, 講談社, 2019)、生物システムを力学系とみなして物理学的手法で解き明かそうとする研究の中でも指摘されている (金子 邦彦, “生命とは何かー複雑系生命科学へ”, 東京大学出版会, 2009; 金子 邦彦, “普遍生物学: 物理に宿る生命、生命の紡ぐ物理”, 東京大学出版会, 2019; 金子 邦彦, 澤井 哲, 高木 拓明, 古澤 力, “細胞の理論生物学: ダイナミクスの視点から”, 東京大学出版会, 2020)。また、2021年にノーベル物理学賞が授与されたスピングラスのような複雑系を扱う物理学などを基礎に急速に発展している深層学習 (田中章詞, 富谷昭夫, 橋本幸士, “ディープラーニングと物理学”, 講談社 2019) も階層性をその特徴に有している。本発明の実施形態に係るプロセス実行システムは、階層型のプロセスネットワークモデルから階層型のプロセスネットワークを生成することが可能である。これにより、本発明の実施形態に係るプロセス実行システムは、複雑系をなすような環境に対してより適応し易いアーキテクチャを有するものと考えられる。

[0055] さらに、深層学習の手法の一つとして、グラフで表現されたデータ構造を扱うGNN (Graph Neural Network) が近年急速に進歩している (Yao Ma, Jiliang Tang, “Deep Learning on Graphs”, Cambridge University Press, 2021; Claudio Stamile, Aldo Marzullo, Enrico Deusebio, “Graph Machine Learning: Take graph data to the next level by applying machine learning techniques and algorithms”, Packt Publishing, 2021)。ネットワークはグラフとして表現可能であり、本発明の実施形態に

係るプロセスネットワークモデルやプロセスネットワークもグラフないし二部グラフとして表現可能である。二部グラフはペトリネットとも呼ばれ、イベントデータからプロセスモデルを抽出するプロセスマイニングでは、プロセスモデルをペトリネットで表現することが多い (Wil M. P. van der Aalst, “Process Mining: Data Science in Action”, Springer, 2016)。

[0056] ビジネスプロセスをプロセスネットワークモデルやプロセスネットワークなどのグラフとして表現することで、GNNなどの深層学習の手法を適用し易くなる。例えば、対象業務のデータから、対象業務に含まれるビジネスプロセスを表現するプロセスネットワークモデルを生成するようなGNNを学習するといったことが考えられる。これにより、これまで手作業で行ってきたビジネスプロセスのモデル化を大幅に自動化できる可能性もある。プロセスネットワークモデルをGNNなどによって生成し、生成されたプロセスネットワークモデルを本発明の実施形態に係るプロセス実行システムに入力してプロセスネットワークを生成し、生成されたプロセスネットワークを当該プロセス実行システムによって実行させることで、対象業務のモデル化からビジネスプロセスの実行までをend to endで自動化できる可能性がある。

[0057] また、異なる企業や業種であっても類似する業務は存在する。一の企業の一の業務に含まれるビジネスプロセスを表現するプロセスネットワークモデルが、当該一の業務に類似する他の企業の他の業務に流用できる可能性がある。本発明の実施形態に係るプロセス実行システムは、プロセスネットワークモデルの実行環境として機能するが、各々のプロセスの実行に必要なアクションに関する実体的な内容はアクション実行システム側で定義される。すなわち、本発明の実施形態に係るプロセス実行システムは、当該プロセス実行システムとアクション実行システムとが疎に結合していることから、上述したプロセスネットワークモデルの流用を容易にする。これにより、本発明

の実施形態に係るプロセス実行システムは、プロセスモデル自体の流通を促進し、プロセスモデルの作成に係る重複投資を軽減し、もって環境負荷の低減にも寄与し得る。

[0058] 以下、各図面を参照して、本発明の実施形態を説明する。

[0059] (第1実施形態)

図1は、本実施形態に係るプロセス実行システム100と、プロセス実行システム100と通信するアクション実行システム50の構成を示す図である。図1に示すように、プロセス実行システム100は、アクション実行システム50との間で通信手段90を介して情報を送受信する。

[0060] 本実施形態に係るプロセス実行システム100は、サーバーマシン上で動作するWebアプリとして構成される。プロセス実行システム100は、Webアプリを構成するためのフレームワークを使用して構成できる。プロセス実行システム100をWebアプリとして構成するためのフレームワークとして、例えばDjango、FlaskおよびRuby on Rails等を使用できる。プロセス実行システム100を実行するサーバーマシンは、オンプレミスのサーバーマシンであってもよく、または商用のクラウドサービスを利用した仮想サーバーマシンであってもよい。

[0061] 本実施形態に係るアクション実行システム50は、サーバーマシン上で動作するWebアプリ、プロセス実行システム100から呼び出されるライブラリ、商用のクラウドサービスが提供するRPAサービス、または商用のRPAアプリ等で構成される。

[0062] アクション実行システム50を、サーバーマシン上で動作するWebアプリとして構成する場合には、アクション実行システム50は、Webアプリを構成するためのフレームワークを使用して構成できる。アクション実行システム50をWebアプリとして構成するためのフレームワークとして、例えば、Django、FlaskおよびRuby on Rails等を使用できる。アクション実行システム50を実行するサーバーマシンは、オンプレミスのサーバーマシンであってもよく、または商用のクラウドサービス

を利用した仮想サーバーマシンであってもよい。

- [0063] アクション実行システム50を、プロセス実行システム100から呼び出されるライブラリとして構成する場合、当該ライブラリは、プロセス実行システム100を構成するライブラリの一部であってよい。アクション実行システム50を構成するライブラリがプロセス実行システム100を構成するライブラリの一部である場合、アクション実行システム50を構成するライブラリは、プロセス実行システム100が動作するサーバーマシン上で実行される。
- [0064] アクション実行システム50を、商用のクラウドサービスが提供するRPAサービスによって構成する場合、アクション実行システム50を構成するRPAサービスとして、例えば、Microsoft Corporationが提供するPower Automateを使用できる。
- [0065] アクション実行システム50を、商用のRPAアプリによって構成する場合、アクション実行システム50を構成するRPAアプリとしては、例えば、Microsoft Corporationが提供するPower Automate for desktopを使用できる。RPAアプリは、クライアントPC上で動作させることができる。
- [0066] 通信手段90は、インターネットやイントラネットなどで構成できる。また、プロセス実行システム100およびアクション実行システム50が、同一のサーバーマシンで動作するオペレーティングシステム(OS)上の異なるタスクとして実行されている場合には、通信手段90は、オペレーティングシステムが備えているタスク間通信でもよい。さらに、アクション実行システム50が、プロセス実行システム100から呼び出されるライブラリとして構成されている場合には、通信手段90は、プロセス実行システム100から当該ライブラリを呼び出す際にオペレーティングシステムが担うデータの送受信手段であってもよい。
- [0067] 図2は、本実施形態に係るプロセス実行システム100のブロック図である。図2に示すように、プロセス実行システム100は、制御部110、記

憶部 120 および通信部 130 を備える。これらの構成要素は、バス 140 を介して互いに接続される。

[0068] 制御部 110 は、CPU (Central Processing Unit)、および RAM (Random Access Memory)、ROM (Read Only Memory) 等のメモリにより構成され、プログラムに従ってプロセス実行システム 100 の各部の制御および演算処理を行う。制御部 110 の機能の詳細については後述する。

[0069] 記憶部 120 は、HDD (Hard Disc Drive)、SSD (Solid State Drive) 等により構成され、各種プログラムおよび各種データを記憶する。記憶部 120 は、後述するプロセスネットワークモデル DM1 を構成するデータやプロセスネットワーク D1 を構成するデータを記憶する。記憶部 120 は、例えば PostgreSQL 等を使用して、データベースサーバーとして機能させることができる。

[0070] 通信部 130 は、ネットワークを介して、外部の装置と通信するためのインターフェース回路 (例えば LAN カード等) である。

[0071] 以下、図 3 から図 10 を参照し、制御部 110 の機能について説明する。図 3 は、プロセス実行システム 100 の制御部 110 の機能を示す機能ブロック図である。図 4 は、プロセス実行システム 100 のプロセスネットワーク生成部 150 の機能ブロック図である。図 5 は、プロセス実行システム 100 のプロセスネットワークモデル入力部 151 が入力を受け付けるプロセスネットワークモデル DM1 のデータ構造を例示する模式図である。図 6 は、プロセス実行システムのプロセスネットワーク生成部 150 によって生成されるプロセスネットワーク D1 のデータ構造を例示する模式図である。図 7 は、プロセス実行システム 100 のプロセスネットワークモデル入力部 151 が入力を受け付けるプロセスネットワークモデル DM1 のプロセスマスターノード DM10 のデータ構造を例示する模式図である。図 8 は、プロセス実行システム 100 のプロセスネットワーク生成部 150 によって生成されるプロセスネットワーク D1 のプロセスノード D10 のデータ構造を例示す

る模式図である。図9は、プロセス実行システム100のプロセスネットワークモデル入力部151が入力を受け付けるプロセスネットワークモデルDM1のプロセスマスタノードDM10のアクション情報マスタDM153のデータ構造を例示する模式図である。図10は、プロセス実行システム100のプロセスネットワーク生成部150によって生成されるプロセスネットワークD1のプロセスノードD10のアクション情報D153のデータ構造を例示する模式図である。

[0072] 図3に示すように、制御部110は、プロセスネットワーク生成部150、プロセスネットワーク状態管理部160、プロセスネットワーク実行部170として機能する。

[0073] 図4に示すように、プロセスネットワーク生成部150は、プロセスネットワークモデルDM1の入力を受け付けるプロセスネットワークモデル入力部151を備え、プロセスネットワークモデル入力部151が入力を受け付けたプロセスネットワークモデルDM1からプロセスネットワークD1を生成する。オブジェクト指向言語との類推で説明すれば、プロセスネットワークモデルDM1はクラスに相当し、プロセスネットワークD1は当該クラスのインスタンスないしオブジェクトに相当する。

[0074] プロセスネットワークモデル入力部151は、プロセスネットワークモデルDM1の入力を受け付けるGUI (Graphical User Interface) を有するように構成してもよいし、プロセスネットワークモデルDM1の入力を受け付けるAPI (Application Programming Interface) を有するように構成してもよい。プロセスネットワークモデルDM1の入力を受け付けるAPIは、例えば、HTTPを使用した通信によってプロセスネットワークモデルDM1の入力を受け付けるように構成できる。また、プロセスネットワークモデルDM1の入力を受け付けるGUIは、例えば、ブラウザからの操作によってプロセスネットワークモデルDM1の入力を受け付けるように構成できる。

[0075] プロセスネットワークモデル入力部151は、プロセスネットワークモデ

ルDM1の入力と合わせて、コンテキストデータの入力を受け付けてもよい。コンテキストデータは、例えば、プロセスネットワークモデルDM1が表現する業務において処理するケースのケースID等のケース属性情報などである。

[0076] プロセスネットワーク生成部150は、プロセスネットワークモデル入力部151が受け付けたプロセスネットワークモデルDM1からプロセスネットワークD1を生成する際に、コンテキストデータを使用することによって、生成されるプロセスネットワークD1の内容を動的に変化させるように構成してもよい。これにより、プロセスネットワーク生成部150は、オブジェクト指向言語との類推で説明すれば、クラスに相当するプロセスネットワークモデルDM1から、当該クラスのオブジェクトに相当するプロセスネットワークD1を生成する際に、生成されるオブジェクトの内容をコンテキストデータに応じて動的に変化させるのと同様のことを実現できる。

[0077] また、プロセスネットワーク生成部150は、プロセスネットワークモデル入力部151が受け付けたコンテキストデータを、生成したプロセスネットワークD1と関連付けて記憶部120に記憶してもよい。これにより、プロセスネットワーク実行部170は、プロセスネットワークD1に含まれるプロセスノードD10を実行する際に、後述するAPIアドレスやAPIデータなどをコンテキストデータに応じて実行時に動的に変化させることによって、アクション実行システム50によって実行されるアクションの内容を実行時に動的に変化させることができる。例えば、プロセスネットワーク実行部170は、コンテキストデータとして記憶部120に記憶されているケース属性情報を使用することによって、アクション実行システム50に送信するAPIデータの内容を動的に変化させることができる。これにより、アクション実行システム50によって生成されるドキュメントの内容等をケース毎に変えることができる。

[0078] 図5に示すように、プロセスネットワークモデルDM1は、プロセスマスタノードDM10を複数含み、プロセスマスタノードDM10同士の間の関

係を表現している。好ましくは、プロセスネットワークモデルDM1は、複数のプロセスマスタノードDM10同士の間の関係を階層的に表現した階層型のプロセスネットワークモデルである。本実施形態では、プロセスネットワークモデルDM1が階層型のプロセスネットワークモデルである場合を例に説明するが、これに限定されない。

[0079] 階層型のプロセスネットワークモデルDM1は、ツリー構造を有する。階層型のプロセスネットワークモデルDM1は、ルートプロセスマスタノードDM10を有する。ルートプロセスマスタノードDM10は、一または複数のプロセスマスタノードDM10を子ノードとして有することができる。プロセスマスタノードDM10は、一または複数のプロセスマスタノードDM10を子ノードとして有することができる。同一の親に属するプロセスマスタノードDM10の数は任意である。階層型のプロセスネットワークモデルDM1は、ルートプロセスマスタノードDM10が属する階層を第0階層、ルートプロセスマスタノードDM10の直属の子のプロセスマスタノードDM10が属する階層を第1階層として、以下、任意の数の階層を有することができる。同一の階層に属するプロセスマスタノードDM10の数は任意である。

[0080] 図6に示すように、プロセスネットワークD1は、プロセスノードD10を複数含み、プロセスノードD10同士の間の関係を表現している。好ましくは、プロセスネットワークD1は、複数のプロセスノードD10同士の間の関係を階層的に表現した階層型のプロセスネットワークである。本実施形態では、プロセスネットワークD1が階層型のプロセスネットワークである場合を例に説明するが、これに限定されない。

[0081] 階層型のプロセスネットワークD1は、ツリー構造を有する。階層型のプロセスネットワークD1は、ルートプロセスノードD10を有する。ルートプロセスノードD10は、一または複数のプロセスノードD10を子ノードとして有することができる。プロセスノードD10は、一または複数のプロセスノードD10を子ノードとして有することができる。同一の親に属する

プロセスノードD10の数は任意である。階層型のプロセスネットワークD1は、ルートプロセスノードD10が属する階層を第0階層、ルートプロセスノードD10の直属の子のプロセスノードが属する階層を第1階層として、以下、任意の数の階層を有することができる。同一の階層に属するプロセスノードD10の数は任意である。

[0082] プロセスネットワーク生成部150は、プロセスネットワークモデルDM1からプロセスネットワークD1を生成する際に、当該プロセスネットワークモデルDM1に含まれるプロセスマスタノードDM10の各々と、当該プロセスネットワークD1に含まれるプロセスノードD10の各々が、一対一に対応するようにプロセスネットワークD1を生成するように構成できる。

[0083] また、プロセスネットワーク生成部150は、プロセスネットワークモデルDM1からプロセスネットワークD1を生成する際に、プロセスネットワークモデル入力部151が入力を受け付けたコンテキストデータに応じて、当該プロセスネットワークモデルDM1に含まれるプロセスマスタノードDM10に対応するプロセスノードD10の生成を制御するように構成してもよい。

[0084] 具体的には、プロセスネットワーク生成部150は、プロセスネットワークモデル入力部151が入力を受け付けたコンテキストデータを特定の種類に分類し、コンテキストデータが一の特定の種類に分類される場合においては一の特定のプロセスマスタノードDM10に対応するプロセスノードD10を生成し、コンテキストデータが他の特定の種類に分類される場合には他の特定のプロセスマスタノードDM10に対応するプロセスノードD10を生成してもよい。すなわち、プロセスネットワーク生成部150は、プロセスネットワークモデルDM1からプロセスネットワークD1を生成する際に、コンテキストデータに応じてプロセスノードD10の生成に使用するプロセスマスタノードDM10を選択的に切り替えるように構成してもよい。例えば、プロセスネットワーク生成部150は、コンテキストデータが一の特

定の顧客名を含んでいる場合には一の特定のプロセスマスタノードDM10に対応するプロセスノードD10を生成し（このとき後述する他の特定のプロセスマスタノードDM10に対応するプロセスノードD10は生成しない）、コンテキストデータが他の特定の顧客名を含んでいる場合には他の特定のプロセスマスタノードDM10に対応するプロセスノードD10を生成する（このとき前述した一の特定のプロセスマスタノードDM10に対応するプロセスノードD10は生成しない）ように構成できる。

[0085] また、プロセスネットワーク生成部150は、プロセスネットワークモデル入力部151が入力を受け付けたコンテキストデータを特定の種類に分類し、コンテキストデータが特定の種類に分類される場合においてのみ特定のプロセスマスタノードDM10に対応するプロセスノードD10を生成する（あるいは生成しない）ように構成してもよい。例えば、プロセスネットワーク生成部150は、コンテキストデータが特定の顧客名を含んでいる場合においてのみ特定のプロセスマスタノードDM10に対応するプロセスノードD10を生成する（あるいは生成しない）ように構成してもよい。

[0086] 図7は、プロセスマスタノードDM10のデータ構造を例示する図である。プロセスマスタノードDM10のデータは、プロセスマスタノード識別子DM11と、親プロセスマスタノード識別子DM12と、プロセス生成情報DM15と、を有する。

[0087] プロセスマスタノード識別子DM11は、プロセスネットワークモデルDM1に含まれるプロセスマスタノードDM10を一意に識別するための識別子である。親プロセスマスタノード識別子DM12は、プロセスマスタノードDM10が属する親のプロセスマスタノードDM10を識別するための識別子である。プロセスマスタノードDM10がルートプロセスマスタノードの場合には、親プロセスマスタノード識別子DM12は値なしに設定できる。

[0088] 図8は、プロセスノードD10のデータ構造を例示する図である。プロセスノードD10のデータは、プロセスノード識別子D11と、親プロセスノ

ード識別子D 1 2と、プロセス状態D 1 3と、プロセス定義情報D 1 5と、を有する。

- [0089] プロセスノード識別子D 1 1は、プロセスネットワークD 1に含まれるプロセスノードD 1 0を一意に識別するための識別子である。親プロセスノード識別子D 1 2は、プロセスノードD 1 0が属する親のプロセスノードD 1 0を識別するための識別子である。プロセスノードD 1 0がルートプロセスノードの場合には、親プロセスノード識別子D 1 2は値なしに設定できる。
- [0090] プロセス状態D 1 3は、プロセスノードD 1 0の状態を表現する。プロセスノードD 1 0の状態は、実行可能状態を含む。好ましくは、プロセスノードD 1 0の状態は完了状態を含む。さらに好ましくは、プロセスノードD 1 0の状態は実行中状態、待機中状態およびエラー終了状態を含む。プロセス状態D 1 3の値は、プロセスノードD 1 0の状態を表現する文字列であってもよく、プロセスノードD 1 0の状態を表現する数値であってもよく、プロセスノードD 1 0の状態を表現するフラグの組み合わせであってもよい。
- [0091] プロセス生成情報DM 1 5は、プロセスネットワーク生成部1 5 0において、当該プロセス生成情報DM 1 5が属するプロセスマスタノードDM 1 0から生成されるプロセスノードD 1 0のプロセス定義情報D 1 5を生成するために使用される。プロセス定義情報D 1 5は、当該プロセス定義情報D 1 5によって定義されたプロセスを実行するのに必要なアクションに関する情報を含む。プロセス生成情報DM 1 5は、各種マスタを含む。プロセス生成情報DM 1 5に含まれる各種マスタの値は、プロセス生成情報DM 1 5によって生成されるプロセス定義情報D 1 5の各項目の値として使用される。
- [0092] プロセスネットワーク生成部1 5 0は、プロセス生成情報DM 1 5に含まれる一のマスタの値がXの場合、当該マスタによって生成されるプロセス定義情報D 1 5の一の項目の値をXとするように構成できる。
- [0093] また、プロセスネットワーク生成部1 5 0は、プロセスネットワークモデル入力部1 5 1が入力を受け付けたプロセスネットワークモデルDM 1からプロセスネットワークD 1を生成する際に、プロセスネットワークモデル入

力部 151 に入力されたコンテキストデータを使用することによって、生成されるプロセス定義情報 D15 を動的に変化させることが可能である。プロセス定義情報 D15 を動的に変化させる方法は特に限定されないが、例えば、Web アプリ作成用のフレームワークである Django で使用されている Django Template Language (DTL) などのテンプレートエンジンの仕組みを使用できる。具体的には、プロセス生成情報 DM15 のマスタの値の中で動的に変化させたい部分にタグを埋め込んでおいて、プロセス定義情報 D15 を生成する際に、プロセスネットワークモデル入力部 151 に入力されたコンテキストデータに応じて当該タグを置換することによってプロセス定義情報 D15 を動的に変化させることが可能である。

[0094] 図 7 を参照して、プロセス生成情報 DM15 は、プロセス番号マスタ DM151 と、プロセス名マスタ DM152 と、アクション情報マスタ DM153 と、を含む。図 8 を参照して、プロセス定義情報 D15 は、プロセス番号 D151 と、プロセス名 D152 と、アクション情報 D153 と、を含む。プロセス番号 D151 はプロセス番号マスタ DM151 によって生成され、プロセス名 D152 はプロセス名マスタ DM152 によって生成され、アクション情報 D153 はアクション情報マスタ DM153 によって生成される。

[0095] プロセス生成情報 DM15 が属するプロセスマスタノード DM10 から生成されるプロセスノード D10 において実行されるアクションが存在しない場合には、当該プロセスマスタノード DM10 はアクション情報マスタ DM153 を含まなくてもよい。例えば、アクションを実行する複数のプロセスマスタノード DM10 を束ねるために配置された親のプロセスマスタノード DM10 の場合、当該親のプロセスマスタノード DM10 において実行されるアクションがない場合がある。プロセスマスタノード DM10 がアクション情報マスタ DM153 を含まない場合、当該プロセスマスタノード DM10 から生成されるプロセスノード D10 はアクション情報 D153 を含まない。

い。

- [0096] プロセス番号マスタDM151の値およびプロセス番号D151の値は、例えば任意の自然数である。プロセスネットワーク生成部150は、プロセス番号マスタDM151の値がN（任意の自然数）の場合、生成されるプロセスノードD10のプロセス番号D151がNとなるように構成できる。プロセス番号D151は、同一階層に属するプロセスノードD10において一意であってもよく、重複していてもよい。また、一部の番号のみが重複していてもよい。例えば、同一階層に属するプロセスノードD10が5つの場合、プロセス番号D151をそれぞれ、100、200、300、300、400というように設定できる。
- [0097] プロセス名マスタDM152およびプロセス名D152の値は、任意の文字列である。プロセスネットワーク生成部150は、プロセス名マスタDM152の値がS（任意の文字列）の場合、生成されるプロセス名D152がSになるように構成できる。
- [0098] アクション情報マスタDM153は、当該アクション情報マスタDM153が属するプロセスマスタノードDM10から生成されるプロセスノードD10において実行されるアクションの内容を定義する。アクションの内容は、アクション実行システム50において実行可能なアクションである限り特に限定されない。
- [0099] Microsoft Corporationが提供するクラウドサービスであるMICROSOFT 365（登録商標）のPower Automateをアクション実行システム50として使用する場合には、アクションの内容は、例えば、ドキュメントの生成、メールの生成、承認依頼の送信、タスクの生成、予定の生成、チャットメッセージの投稿などである。ドキュメントは、例えば、メールや請求書などである。Power Automateを使用することで、各種ドキュメントを生成することが可能である。ドキュメントのファイル形式としては、例えば、Microsoft CorporationのOfficeのWordで使用可能なファイル形式、

EXCEL（登録商標）で使用可能なファイル形式、OUTLOOK（登録商標）で使用可能なファイル形式、POWERPOINT（登録商標）で使用可能なファイル形式などである。承認依頼の送信は、Power AutomateからMICROSOFT 365（登録商標）の承認依頼アプリに承認依頼を送信することによって実行できる。タスクの生成は、Power AutomateからMICROSOFT 365（登録商標）のPlannerにタスクを登録することによって実行できる。予定の生成は、MICROSOFT 365（登録商標）のOUTLOOK（登録商標）に予定を生成することによって実行できる。チャットメッセージの投稿は、Power AutomateからMICROSOFT 365（登録商標）のTeamsの特定のユーザーとのチャットへのメッセージの投稿、特定のチャンネルへのメッセージの投稿、特定のチャットグループへのメッセージの投稿などによって実行できる。

[0100] 図9は、アクション情報マスタDM153のデータ構造を例示する図である。アクション情報マスタDM153は、一または複数のアクションアイテムマスタDM1531を含む。

[0101] 図10は、アクション情報D153のデータ構造を例示する図である。アクション情報D153は、一または複数のアクションアイテムD1531を含む。

[0102] 図9を参照して、アクションアイテムマスタDM1531は、アクションアイテム番号マスタと、APIアドレスマスタと、APIデータマスタと、を有する。図10を参照して、アクションアイテムは、アクションアイテム番号と、APIアドレスと、APIデータと、を有する。アクションアイテム番号はアクションアイテム番号マスタによって生成され、APIアドレスはAPIアドレスマスタによって生成され、APIデータはAPIデータマスタによって生成される。

[0103] アクションアイテム番号マスタの値およびアクションアイテム番号の値は、例えば任意の自然数である。プロセスネットワーク生成部150は、アク

ションアイテム番号マスタの値がN（任意の自然数）の場合、生成されるプロセスノードD10のアクションアイテム番号がNになるように構成できる。アクションアイテム番号は、一のアクション情報D153に属する一または複数のアクションアイテムにおいて一意であってもよく、重複していてもよい。また、一部のアクションアイテム番号のみが重複していてもよい。例えば、一のアクション情報D153に属するアクションアイテムが5つの場合、アクションアイテム番号をそれぞれ、1、2、3、3、4というように設定できる。

[0104] APIアドレスは、当該APIアドレスが属するアクションアイテムによって定義されるアクションを実行するために使用する。プロセスネットワーク実行部170は、実行可能なプロセスノードD10に含まれるアクションを実行する際に、アクション実行システム50に対してAPIアドレスで指定されたAPI経由でアクション実行要求を送信して、アクション実行システム50に当該アクションを実行させる。

[0105] APIアドレスマスタの値およびAPIアドレスの値は、文字列である。プロセスネットワーク生成部150は、APIアドレスマスタの値がS（任意の文字列）の場合、生成されるAPIアドレスがSになるように構成できる。プロセスネットワーク実行部170がアクション実行システム50のAPIにHTTPリクエストを送信してアクション実行システム50にアクションを実行させる場合には、APIアドレスはURLである。例えば、アクション実行システム50がPower Automateである場合には、APIアドレスは、Power Automateのフロー起動用のAPIのURLである。また、アクション実行システム50が、Django等のWebフレームワークによって構築されたオンプレミスサーバーもしくは仮想サーバー上で動作するWebアプリの場合には、APIアドレスは、当該Webアプリのアクション起動用のURLである。

[0106] APIデータマスタの値およびAPIデータの値は、例えば任意の文字列である。プロセスネットワーク生成部150は、APIデータマスタの値が

S（任意の文字列）の場合、生成されるAPIデータがSになるように構成できる。プロセスネットワーク実行部170がアクション実行システム50のAPIにHTTPリクエストを送信してアクション実行システム50にアクションを実行させる場合には、APIデータはHTTPリクエストにおけるボディである。例えば、アクション実行システム50がPower Automateである場合には、APIデータは、Power Automateのフロー起動用のAPIにPOSTするボディである。

[0107] アクション情報マスタDM153のデータ形式は特に限定されないが、例えばJSON形式を使用できる。JSON形式を使用することによって、一または複数のアクションアイテムマスタDM1531を含むアクション情報マスタDM153を一つの文字列で表現できる。これにより、プロセスネットワーク生成部150がプロセスネットワークモデルDM1からプロセスネットワークD1を生成する際に、プロセスネットワークモデル入力部151が受け付けたコンテキストデータに応じてアクション情報マスタDM153からアクション情報D153を動的に変化させる処理が容易になる。例えば、プロセス定義情報D15を動的に変化させる方法として上述したようにWebアプリ作成用のフレームワーク（Django等）で使用されているようなテンプレートエンジンの仕組みを使用した場合、アクション情報マスタDM153に含まれるタグをコンテキストデータに応じて変換する際に、当該タグを変換後の値で一括して置換することができる。また、JSON形式のデータは、例えばMicrosoft Corporationが提供するVS CODE（登録商標）等のエディタを使用して効率よく作成や編集を行える。そのため、アクション情報マスタDM153の編集やメンテナンスを業務担当者が容易に行える。また、プロセスネットワークモデルDM1をGNN等の機械学習によって学習させる際に、アクションに関する情報が一つの文字列として表現されているため学習させ易い。

[0108] アクション情報D153のデータ形式は特に限定されないが、例えばJSON形式を使用できる。JSON形式を使用することによって、一または複

数のアクションアイテムを含むアクション情報D 1 5 3を一つの文字列で表現できる。また、プロセスネットワークD 1のデータを使用してプロセスネットワークモデルDM 1をGNN等の機械学習によって学習させる際に、アクションに関する情報が一つの文字列として表現されているため学習させ易い。

[0109] プロセスネットワーク状態管理部1 6 0は、プロセスネットワークD 1に含まれるプロセスノードD 1 0の状態と、所定のプロセス状態遷移規則と、に基づいてプロセスノードD 1 0の状態遷移処理を行う。具体的には、プロセスネットワーク状態管理部1 6 0は、プロセスネットワークD 1に含まれるプロセスノードD 1 0の状態と、所定のプロセス状態遷移規則と、に基づいて、特定のプロセスノードD 1 0を実行可能にする。プロセスネットワーク状態管理部1 6 0の動作の詳細は、図1 2から図1 4を参照して後述する。

[0110] プロセスネットワーク実行部1 7 0は、プロセスネットワークD 1に含まれる実行可能なプロセスノードD 1 0が有するプロセス定義情報D 1 5によって定義されたプロセスを実行する。プロセスネットワーク実行部1 7 0は、例えば、実行可能なプロセスノードD 1 0をキューに保存しておいて、実行可能なプロセスノードD 1 0のプロセスを順次または並列に実行できる。

[0111] プロセスネットワーク実行部1 7 0は、実行可能なプロセスノードD 1 0のプロセス定義情報D 1 5が含むアクション情報D 1 5 3に基づいて当該アクションの実行要求をアクション実行システム5 0に送信することによって、当該プロセス定義情報D 1 5によって定義されたプロセスを実行する。具体的には、プロセスネットワーク実行部1 7 0は、実行可能なプロセスノードD 1 0のプロセス定義情報D 1 5に含まれるアクションアイテムの内容に基づいて、アクション実行システム5 0に対してアクションの実行要求を送信する。より具体的には、プロセスネットワーク実行部1 7 0は、アクションアイテムに含まれるAPIアドレスに対してAPIデータを送信することでアクション実行システム5 0にアクションを実行させる。例えば、アクシ

ョン実行システム50がHTTPによるAPI経由でのアクション実行要求を受け付ける場合には、プロセスネットワーク実行部170は、アクションアイテムに含まれるAPIアドレスに対してAPIデータをボディとするHTTPのPOSTを行う。

[0112] 次に図11から図15を参照し、プロセス実行システム100の動作を説明する。図11は、プロセス実行システム100において実行される処理の手順を表すフローチャートである。図12は、プロセス実行システム100において実行されるプロセスノードの状態遷移処理の手順を表すフローチャートである。図13は、プロセス実行システム100において実行される親プロセスノードの配下のプロセスノード群に対する状態遷移処理の手順を表すフローチャートである。図14は、プロセス実行システム100において実行される最小のプロセス番号D151を有するプロセスノード群に対する状態遷移処理の手順を表すフローチャートである。図15は、プロセス実行システム100において実行される実行可能なプロセスノードD10の実行処理の手順を表すフローチャートである。図11から図15のフローチャートに示す処理は、プログラムに従い、プロセス実行システム100の制御部110により実行される。

[0113] 図11に示すように、プロセス実行システム100の制御部110により実行されるプログラムは、プロセスネットワークモデルDM1の入力を受け付けるステップS11と、プロセスネットワークD1を生成するステップS12と、プロセスノードD10の状態遷移処理を行うステップS13と、実行可能なプロセスノードD10の実行処理を行うステップS14と、プロセスネットワークD1に含まれる全てのプロセスノードD10が完了状態になっているか判断するステップS15と、を有する。

[0114] (ステップS11)

プロセスネットワークモデル入力部151は、プロセスの内容を定義したプロセス定義情報D15を生成するために使用されるプロセス生成情報DM15を有するプロセスマスタノードDM10を複数含み、当該複数のプロセ

スマスタノードDM10によって生成されるプロセスノードD10同士の間
の関係を表現したプロセスネットワークモデルDM1の入力を受け付ける。
プロセスネットワークモデル入力部151は、プロセスネットワークモデル
DM1の入力を受け付ける際に、コンテキストデータの入力を受け付けても
よい。

[0115] (ステップS12)

プロセスネットワーク生成部150は、プロセスネットワークモデル入力
部151が入力を受け付けたプロセスネットワークモデルDM1から、プロ
セス定義情報D15を有するプロセスノードD10を複数含むプロセスネッ
トワークD1を生成する。

[0116] (ステップS13)

プロセスネットワーク状態管理部160は、プロセスネットワークD1に
含まれるプロセスノードD10の状態と、所定のプロセス状態遷移規則と、
に基づいてプロセスノードD10の状態遷移処理を行う。

[0117] 具体的には、プロセスネットワーク状態管理部160は、プロセスネッ
トワークD1に含まれるプロセスノードD10の状態と、所定のプロセス状態
遷移規則と、に基づいて、特定のプロセスノードD10を実行可能にする。
プロセスネットワーク状態管理部160は、プロセスネットワークD1に含
まれる複数のプロセスノードD10の状態の組み合わせに基づいて特定のプ
ロセスノードD10を実行可能にするように構成できる。プロセスネッ
トワーク状態管理部160は、プロセスネットワークD1に含まれるプロセスノ
ードD10のうち、アクション情報D153を有するプロセスノードD10
の各々を少なくとも1回は実行可能にするように構成できる。

[0118] プロセスネットワーク状態管理部160は、プロセスネットワークD1が
階層型のプロセスネットワークである場合、プロセスネットワークD1に含
まれる異なる階層に属する複数のプロセスノードD10の状態の組み合わせ
に応じて特定のプロセスを実行可能にするように構成できる。プロセスネッ
トワーク状態管理部160は、プロセスネットワークD1が階層型のプロセ

スネットワークである場合、深い階層に属するプロセスノードD 1 0の方が浅い階層に属するプロセスノードD 1 0よりも優先して実行可能になるように構成できる。プロセスネットワーク状態管理部1 6 0は、プロセスネットワークD 1が階層型のプロセスネットワークである場合、子プロセスノードD 1 0の実行が完了した後に当該子プロセスノードD 1 0の親のプロセスノードD 1 0が実行可能になるように構成できる。

[0119] (ステップS 1 4)

プロセスネットワーク実行部1 7 0は、プロセスネットワークD 1に含まれる実行可能なプロセスノードD 1 0の実行処理を行う。

[0120] (ステップS 1 5)

プロセスネットワーク実行部1 7 0は、プロセスネットワークD 1に含まれる全てのプロセスノードD 1 0が完了状態になっているか判断する。プロセスネットワークD 1に含まれる全てのプロセスノードD 1 0が完了状態になっていない場合には、ステップS 1 3に戻る。プロセスネットワークD 1に含まれる全てのプロセスノードD 1 0が完了状態になっている場合には、処理を終了する。

[0121] 図1 2から図1 4を参照して、ステップS 1 3において行われるプロセスノードD 1 0の状態遷移処理について説明する。

[0122] 図1 2を参照して、ステップS 1 3において行われるプロセスノードD 1 0の状態遷移処理は、ルートプロセスノードD 1 0を親プロセスノードに設定するステップS 1 3 1と、親プロセスノードの配下のプロセスノード群に対する状態遷移処理を実行するステップS 1 3 2と、を有する。

[0123] 図1 3を参照して、親プロセスノードの配下のプロセスノード群に対する状態遷移処理は、親プロセスノードの子プロセスノードの中で未完了のプロセスノードD 1 0を抽出するステップS 1 3 2 1と、未完了のプロセスノードD 1 0が存在するか否か判断するステップS 1 3 2 2と、未完了のプロセスノードD 1 0の中から最小のプロセス番号D 1 5 1を有するプロセスノード群を抽出するステップS 1 3 2 3と、最小のプロセス番号D 1 5 1を有

するプロセスノード群に対する状態遷移処理を実行するステップS 1 3 2 4と、親プロセスノードがアクション情報D 1 5 3を有するか否か判断するステップS 1 3 2 5と、親プロセスノードを実行可能状態にするステップS 1 3 2 6と、親プロセスノードを完了状態にするステップS 1 3 2 7と、を有する。

[0124] (ステップS 1 3 2 1)

プロセスネットワーク状態管理部1 6 0は、親プロセスノードの子のプロセスノードの中で未完了のプロセスノードD 1 0を抽出する。親プロセスノードの子のプロセスノードとは、親プロセスノードD 1 0の直属の子のプロセスノードD 1 0を意味し、親プロセスノードD 1 0の直属の子のプロセスノードD 1 0の子のプロセスノードD 1 0（親プロセスノードD 1 0の孫のプロセスノードD 1 0）等は含まない。

[0125] (ステップS 1 3 2 2)

プロセスネットワーク状態管理部1 6 0は、ステップS 1 3 2 1において抽出した未完了のプロセスノードD 1 0が存在するか否かを判断する。未完了のプロセスノードD 1 0が存在する場合には、ステップS 1 3 2 3に進む。未完了のプロセスノードD 1 0が存在しない場合には、ステップS 1 3 2 5に進む。

[0126] (ステップS 1 3 2 3)

プロセスネットワーク状態管理部1 6 0は、ステップS 1 3 2 1で抽出した未完了のプロセスノードD 1 0の中から、最小のプロセス番号D 1 5 1を有するプロセスノード群を抽出する。最小のプロセス番号D 1 5 1を有するプロセスノード群とは、最小のプロセス番号D 1 5 1を有するプロセスノードD 1 0によって構成される集合を意味する。

[0127] (ステップS 1 3 2 4)

プロセスネットワーク状態管理部1 6 0は、ステップS 1 3 2 3で抽出した最小のプロセス番号D 1 5 1を有するプロセスノード群に対する状態遷移処理を実行する。

[0128] (ステップS 1 3 2 5)

プロセスネットワーク状態管理部160は、親プロセスノードがアクション情報D 1 5 3を有するか否か判断する。親プロセスノードがアクション情報D 1 5 3を有する場合には、ステップS 1 3 2 6に進む。アクション情報D 1 5 3を有していない場合には、ステップS 1 3 2 7に進む。

[0129] (ステップS 1 3 2 6)

プロセスネットワーク状態管理部160は、親プロセスノードを実行可能状態にする。

[0130] (ステップS 1 3 2 7)

プロセスネットワーク状態管理部160は、親プロセスノードを完了状態にする。

[0131] 図14を参照して、最小のプロセス番号D 1 5 1を有するプロセスノード群に対する状態遷移処理は、最小のプロセス番号D 1 5 1を有するプロセスノードD 1 0が子プロセスノードD 1 0を有するか否か判断するステップS 1 3 2 4 1と、最小のプロセス番号D 1 5 1を有するプロセスノードD 1 0を親プロセスノードに設定するステップS 1 3 2 4 2と、親プロセスノードの配下のプロセスノード群に対する状態遷移処理を実行するステップS 1 3 2 4 3と、最小のプロセス番号D 1 5 1を有するプロセスノードD 1 0がアクション情報D 1 5 3を有するか判断するステップS 1 3 2 4 4と、最小のプロセス番号D 1 5 1を有するプロセスノードD 1 0を実行可能状態にするステップS 1 3 2 4 5と、最小のプロセス番号D 1 5 1を有するプロセスノードD 1 0を完了状態にするステップS 1 3 2 4 6と、最小のプロセス番号D 1 5 1を有するプロセスノードD 1 0全てについて処理したか否か判断するステップS 1 3 2 4 7と、を有する。

[0132] (ステップS 1 3 2 4 1)

プロセスネットワーク状態管理部160は、最小のプロセス番号D 1 5 1を有するプロセスノードD 1 0が子プロセスノードを有するか否か判断する。最小のプロセス番号D 1 5 1を有するプロセスノードD 1 0が子のプロ

セスノードを有する場合には、ステップS 1 3 2 4 2に進む。最小のプロセス番号D 1 5 1を有するプロセスノードD 1 0が子のプロセスノードを有していない場合には、ステップS 1 3 2 4 4に進む。

[0133] (ステップS 1 3 2 4 2)

プロセスネットワーク状態管理部1 6 0は、最小のプロセス番号D 1 5 1を有するプロセスノードD 1 0を新たな親プロセスノードに設定する。

[0134] (ステップS 1 3 2 4 3)

プロセスネットワーク状態管理部1 6 0は、親プロセスノードの配下のプロセスノード群に対する状態遷移処理を実行する。

[0135] (ステップS 1 3 2 4 4)

プロセスネットワーク状態管理部1 6 0は、最小のプロセス番号D 1 5 1を有するプロセスノードD 1 0がアクション情報D 1 5 3を有するか否かを判断する。アクション情報D 1 5 3を有している場合には、ステップS 1 3 2 4 5に進む。アクション情報D 1 5 3を有していない場合には、ステップS 1 3 2 4 6に進む。

[0136] (ステップS 1 3 2 4 5)

プロセスネットワーク状態管理部1 6 0は、最小のプロセス番号D 1 5 1を有するプロセスノードD 1 0を実行可能状態にする。

[0137] (ステップS 1 3 2 4 6)

プロセスネットワーク状態管理部1 6 0は、最小のプロセス番号D 1 5 1を有するプロセスノードD 1 0を完了状態にする。

[0138] (ステップS 1 3 2 4 7)

プロセスネットワーク状態管理部1 6 0は、最小のプロセス番号D 1 5 1を有するプロセスノード群に含まれるプロセスノードD 1 0を全て処理したか否かを判断する。最小のプロセス番号D 1 5 1を有するプロセスノード群に含まれるプロセスノードD 1 0を全て処理していない場合には、ステップS 1 3 2 4 1に戻る。最小のプロセス番号D 1 5 1を有するプロセスノード群に含まれるプロセスノードD 1 0を全て処理した場合には処理を終了する。

[0139] 図 15 を参照して、ステップ S 14 において行われる実行可能なプロセスノード D 10 の実行処理は、実行可能なプロセスノード D 10 が存在するか否か判断するステップ S 141 と、実行可能なプロセスノード D 10 を実行するステップ S 142 と、実行を完了したプロセスノード D 10 の状態を完了状態にするステップ S 143 と、を有する。

[0140] (ステップ S 141)

プロセスネットワーク実行部 170 は、プロセスノード D 10 が実行可能状態か否かを判断する。プロセスノード D 10 が実行可能状態か否かの判断は、プロセスノード D 10 に含まれるプロセス状態 D 13 の値を調べることによって判断する。

[0141] (ステップ S 142)

プロセスネットワーク実行部 170 は、実行可能なプロセスノード D 10 のプロセス定義情報 D 15 が含むアクションに関する情報に基づいて当該アクションの実行要求をアクション実行システム 50 に送信することによって、当該プロセス定義情報 D 15 によって定義されたプロセスを実行する。具体的には、プロセスネットワーク実行部 170 は、実行可能なプロセスノード D 10 のプロセス定義情報 D 15 に含まれるアクションアイテムの内容に基づいて、アクション実行システム 50 に対してアクションの実行要求を送信する。プロセスネットワーク実行部 170 は、アクション情報 D 153 が複数のアクションアイテムを含む場合には、アクションアイテム番号が小さい順にアクションアイテムを実行してもよい。

[0142] (ステップ S 143)

プロセスネットワーク実行部 170 は、実行を完了したプロセスノード D 10 のプロセス状態 D 13 を完了状態にする。プロセス状態 D 13 を完了状態にする方法は特に限定されないが、例えば、プロセスネットワーク実行部 170 は、記憶部 120 に記憶されているプロセスネットワーク D 1 のプロセスノード D 10 のプロセス状態 D 13 を完了状態にするように記憶部 120 に対して要求することによって、実行を完了したプロセスノード D 10 の

プロセス状態D 1 3を完了状態にすることができる。記憶部1 2 0に対する要求は、記憶部1 2 0がデータベースサーバーとして機能している場合には、SQLクエリの送信によって行うことができる。

[0143] 図1 6および図1 7を参照して、図1 2から図1 4を用いて説明したプロセスノードD 1 0の状態遷移処理を行うステップS 1 3の動作例を説明する。

[0144] 図1 6は、プロセス実行システム1 0 0において実行されるプロセスノードD 1 0の状態遷移処理を行うステップS 1 3の動作例について説明するために使用するプロセスネットワークD 1のデータ構造を示す模式図である。図1 6に示すプロセスネットワークD 1において、ルートプロセスノードL 0 1は、プロセスノードL 1 1、L 1 2およびL 1 3を子のプロセスノードとして有する。プロセスノードL 1 1は、プロセスノードL 2 1、L 2 2およびL 2 3を子のプロセスノードとして有する。プロセスノードL 1 2は、プロセスノードL 2 4およびL 2 5を子のプロセスノードとして有する。図2 0に示すプロセスネットワークD 1に含まれるプロセスノードのうち、灰色で着色したプロセスノードL 2 1、L 2 2、L 2 3、L 2 4、L 2 5およびL 1 3はアクション情報D 1 5 3を有し、灰色で着色されていないプロセスノードL 0 1、L 1 1およびL 1 2はアクション情報D 1 5 3を有していない。各ノード内の数字はプロセス番号を表す。

[0145] 図1 7は、プロセス実行システム1 0 0において実行されるプロセスノードD 1 0の状態遷移処理を行うステップS 1 3の動作例について説明するために使用するプロセスノードの状態遷移を表す図である。プロセスネットワーク状態管理部1 6 0によるプロセスノードD 1 0の状態遷移処理は、一のプロセスネットワークD 1に含まれる全てのプロセスノードD 1 0が完了状態になるまで繰り返し適用される(図1 1参照)。図1 7は、状態遷移処理を行うことによってプロセスノードD 1 0の状態がどのように遷移するかを示す。なお、以下の説明では、一の状態遷移処理から次の状態遷移処理を実行するまでの間に、当該一の状態遷移処理で実行可能状態になったプロセス

ノードD 1 0は、プロセスネットワーク実行部1 7 0によって実行されて完了状態になっているものとする。

[0146] 図1 2を参照して、まず、ステップS 1 3 1においてルートプロセスノードL 0 1を親プロセスノードに設定する。次に、ステップS 1 3 2において、親プロセスノードL 0 1の配下のプロセスノード群に対する状態遷移処理を実行する。

[0147] 図1 3を参照して、親プロセスノードL 0 1の配下のプロセスノード群に対する状態遷移処理では、ステップS 1 3 2 1において、親プロセスノードL 0 1の子のプロセスノードL 1 1、L 1 2およびL 1 3の中で、未完了のプロセスノードL 1 1、L 1 2およびL 1 3を抽出する。ステップS 1 3 2 1において未完了の子のプロセスノードL 1 1、L 1 2およびL 1 3が抽出されたので、ステップS 1 3 2 2の判断はY e sになる。次に、ステップS 1 3 2 3において、未完了のプロセスノードL 1 1、L 1 2およびL 1 3の中から最小のプロセス番号1 0 0を有するプロセスノード群 {L 1 1, L 1 2} を抽出する。ステップS 1 3 2 4において、最小のプロセス番号1 0 0を有するプロセスノード群 {L 1 1, L 1 2} に対する状態遷移処理を実行する。

[0148] 図1 4を参照して、最小のプロセス番号1 0 0を有するプロセスノード群 {L 1 1, L 1 2} に対する状態遷移処理は、最小のプロセス番号1 0 0を有するプロセスノード群 {L 1 1, L 1 2} に含まれるプロセスノードL 1 1およびL 1 2の各々について行われる。

[0149] まず、プロセスノードL 1 1の配下のプロセスノード群に対する状態遷移処理について説明する。プロセスノードL 1 1は子のプロセスノードL 2 1、L 2 2およびL 2 3を有するからステップS 1 3 2 4 1の判断はY e sとなり、ステップS 1 3 2 4 2に進む。ステップS 1 3 2 4 2ではプロセスノードL 1 1を新たな親プロセスノードに設定する。ステップS 1 3 2 4 3において、親プロセスノードL 1 1の配下のプロセスノード群に対する状態遷移処理を実行する。図1 3を参照して、親プロセスノードL 1 1の配下のプ

プロセスノード群に対する状態遷移処理では、ステップS 1 3 2 1において、親のプロセスノードL 1 1の子のプロセスノードの中から未完了のプロセスノードL 2 1、L 2 2およびL 2 3を抽出する。未完了のプロセスノードL 2 1、L 2 2およびL 2 3が存在するから、ステップS 1 3 2 2の判断はY e sになり、ステップS 1 3 2 3に進む。ステップS 1 3 2 3において、未完了の子のプロセスノードL 2 1、L 2 2およびL 2 3の中から、最小のプロセス番号1 0 0を有するプロセスノード群 {L 2 1} を抽出する。ステップS 1 3 2 4において、最小のプロセス番号1 0 0を有するプロセスノード群 {L 2 1} に対する状態遷移処理を実行する。図1 4を参照して、最小のプロセス番号1 0 0を有するプロセスノード群 {L 2 1} に対する状態遷移処理では、プロセスノードL 2 1が子のプロセスノードを有していないから、ステップS 1 3 2 4 1の判断はN oとなり、ステップS 1 3 2 4 4に進む。プロセスノードL 2 1はアクション情報D 1 5 3を有するからステップS 1 3 2 4 4の判断はY e sとなり、ステップS 1 3 2 4 5に進む。ステップS 1 3 2 4 5において、プロセスノードL 2 1を実行可能状態にして、プロセスノードL 1 1の配下のプロセスノード群に対する状態遷移処理は終了する。

[0150] プロセスノードL 1 2配下のプロセスノード群に対する状態遷移処理についても同様に行われ、プロセスノードL 2 4を実行可能状態にして処理が終了する。

[0151] 図1 4を参照して、プロセスノードL 1 1の配下のプロセスノード群に対する状態遷移処理およびプロセスノードL 1 2配下のプロセスノード群に対する状態遷移処理を完了すると、S 1 3 2 4 7の判断がY e sになって、最小のプロセス番号1 0 0を有するプロセスノード群 {L 1 1, L 1 2} に対する状態遷移処理を完了する。これにより、1回目の状態遷移処理が完了する。

[0152] 次に2回目の状態遷移処理を行うと、1回目の状態遷移処理と同様に処理が行われてプロセスノードL 2 2およびプロセスノードL 2 5が実行可能状

態になる。

[0153] 次に3回目の状態遷移処理を行うと、図12のステップS131、S132、図13のステップS1321、S1322、S1323の順に進む。ステップS1323において最小のプロセス番号100を有するプロセスノード群{L11, L12}を抽出し、ステップS1324に進む。ステップS1324において、最小のプロセス番号100を有するプロセスノード群{L11, L12}に対する状態遷移処理を実行する。

[0154] 図14を参照して、プロセスノードL11の配下のプロセスノード群に対する状態遷移処理では、ステップS13241、S13242、S13243の順に進む。ステップS13243において、親プロセスノードL11の配下のプロセスノード群に対する状態遷移処理を実行する。図13を参照して、親プロセスノードL11の配下のプロセスノード群に対する状態遷移処理では、ステップS1321、S1322、S1323、S1324の順に進む。ステップS1324において、最小のプロセス番号300を有するプロセスノード群{L23}に対する状態遷移処理を実行する。図14を参照して、最小のプロセス番号300を有するプロセスノード群{L23}に対する状態遷移処理では、ステップS13241、S13244、S13245の順に進み、プロセスノードL23を実行可能状態にして処理を終了する。

[0155] 図14を参照して、プロセスノードL12の配下のプロセスノード群に対する状態遷移処理では、S13241、S13242、S13243の順に進む。ステップS13243において、親プロセスノードL12の配下のプロセスノード群に対する状態遷移処理を実行する。図13を参照して、親プロセスノードL12の配下のプロセスノード群に対する状態遷移処理では、ステップS1321、S1322、S1325、S1327の順に進み、プロセスノードL12を完了状態にして処理を終了する。

[0156] 次に4回目の状態遷移処理を行うと、図12のステップS131、S132、図13のステップS1321、S1322、S1323の順に進む。ス

ステップS 1 3 2 3において、最小のプロセス番号1 0 0を有するプロセスノード群 {L 1 1} が抽出され、ステップS 1 3 2 4において、最小のプロセス番号1 0 0を有するプロセスノード群 {L 1 1} に対する状態遷移処理を実行する。図1 4を参照して、最小のプロセス番号1 0 0を有するプロセスノードL 1 1に対する状態遷移処理では、ステップS 1 3 2 4 1、S 1 3 2 4 2、S 1 3 2 4 3の順に進む。ステップS 1 3 2 4 3において、親プロセスノードL 1 1の配下のプロセスノード群に対する状態遷移処理を実行する。図1 3を参照して、親プロセスノードL 1 1の配下のプロセスノード群に対する状態遷移処理では、ステップS 1 3 2 1、S 1 3 2 2、S 1 3 2 5、S 1 3 2 7の順に進み、プロセスノードL 1 1を完了状態にして処理を終了する。

[0157] 次に5回目の状態遷移処理を行うと、図1 2のステップS 1 3 1、S 1 3 2、図1 3のステップS 1 3 2 1、S 1 3 2 2、S 1 3 2 3の順に進む。ステップS 1 3 2 3において最小のプロセス番号2 0 0を有するプロセスノード群 {L 1 3} が抽出され、ステップS 1 3 2 4に進む。ステップS 1 3 2 4において、最小のプロセス番号2 0 0を有するプロセスノード群 {L 1 3} に対する状態遷移処理を実行する。図1 4を参照して、最小のプロセス番号2 0 0を有するプロセスノードL 1 3に対する状態遷移処理では、ステップS 1 3 2 4 1、S 1 3 2 4 4、S 1 3 2 4 5の順に進み、プロセスノードL 1 3を実行可能状態にして処理を終了する。

[0158] 次に6回目の状態遷移処理を行うと、図1 2のステップS 1 3 1、S 1 3 2、図1 3のステップS 1 3 2 1、S 1 3 2 2、S 1 3 2 5、S 1 3 2 7の順に進む。ステップS 1 3 2 7において、親プロセスノードL 0 1（ルートプロセスノード）を完了状態にして処理を終了する。これにより、プロセスネットワークD 1に含まれる全てのプロセスノードD 1 0の状態が完了状態になる。

[0159] （作用効果）

本実施形態に係るプロセス実行システム1 0 0は、プロセスの内容を定義

したプロセス定義情報D15を生成するために使用されるプロセス生成情報DM15を有するプロセスマスタノードDM10を複数含み、当該複数のプロセスマスタノードDM10によって生成されるプロセスノードD10同士の間 の関係を表現したプロセスネットワークモデルDM1から、プロセス定義情報D15を有するプロセスノードD10を複数含むプロセスネットワークD1を生成するプロセスネットワーク生成部150と、プロセスネットワークD1に含まれるプロセスノードD10の状態と、所定のプロセス状態遷移規則と、に基づいて、特定のプロセスノードD10を実行可能にするプロセスネットワーク状態管理部160と、プロセスネットワークD1に含まれる実行可能なプロセスノードD10が有するプロセス定義情報D15によって定義されたプロセスを実行するプロセスネットワーク実行部170と、を有し、プロセスネットワーク生成部150は、プロセスネットワークモデルDM1の入力を受け付けるプロセスネットワークモデル入力部151を備え、当該プロセスネットワークモデル入力部151が入力を受け付けたプロセスネットワークモデルDM1からプロセスネットワークD1を生成し、プロセス定義情報D15は、当該プロセス定義情報D15によって定義されたプロセスを実行するのに必要なアクションに関する情報D153を含み、プロセスネットワーク実行部170は、要求に応じて所定のアクションを実行するアクション実行システム50に対して、実行可能なプロセスノードD10のプロセス定義情報D15が含むアクションに関する情報D153に基づいて当該アクションの実行要求を送信することによって、当該プロセス定義情報D15によって定義されたプロセスを実行する。

[0160] これにより、本実施形態に係るプロセス実行システム100は、プロセスネットワークモデルDM1の実行環境として機能し、プロセスネットワークモデルDM1を変更することによって実行結果も変化させることができるため、業務内容の変更等に柔軟に対応できる。

[0161] また、プロセスの実行に必要なアクションは、プロセス実行システム100から独立したアクション実行システム50によって実行される。これによ

り、プロセスの実行に必要なアクションの実体的な内容は、アクション実行システム50側に定義すればよく、プロセスネットワークモデルDM1には、アクション実行システム50に対するアクションの実行要求の方法を定義すればよい。そのため、業務担当者自らがプロセスネットワークモデルDM1の作成や修正等を行い易くなるから、より柔軟に業務内容の変更等に対応できる。

[0162] また、プロセスの実行に必要なアクションの実体的な内容をアクション実行システム50側に定義することで、異なる業務間で類似するアクションを再利用し易くなる。すなわち、一の業務を表現するプロセスネットワークモデルで使用するアクションを再利用することで、類似する他の業務をプロセスネットワークモデルによって素早く表現して実行することができる。そのため、本実施形態に係るプロセス実行システム100によれば、多様な業務に対して素早く適応することが容易になる。

[0163] さらに、業務を構成するビジネスプロセスが、プロセスネットワークモデルDM1として表現されるため、プロセス同士の関係が把握し易い。これにより、一部のプロセスの実行結果や変更内容が他のプロセスに予期しない影響を与えることを未然に防ぎ易くなる。また、プロセスネットワークモデルDM1がその実行環境（プロセス実行システム100）から分離しているため、プロセスネットワークモデルDM1を参照することによって、プロセスネットワークモデルDM1に基づいて実行される業務の内容を評価し易くなる。その結果、安定した業務運営をし易くなる。

[0164] 以上のことから、本実施形態に係るプロセス実行システム100によれば、可塑性と安定性とを両立させることによって変化の速度が速いビジネス環境への適応度が高いプロセス実行システム100を提供できる。

[0165] また、本実施形態に係るプロセス実行システム100において、プロセスネットワークモデル入力部151は、複数のプロセスマスタノードDM10によって生成されるプロセスノードD10同士の間の関係を階層的に表現した階層型のプロセスネットワークモデルDM1の入力を受け付け、プロセス

ネットワーク生成部１５０は、プロセスネットワークモデル入力部１５１が受け付けた階層型のプロセスネットワークモデルＤＭ１に対応する階層型のプロセスネットワークＤ１を生成する。

[0166] 本実施形態に係るプロセス実行システム１００によれば、プロセスネットワークモデルＤＭ１は階層的に表現される。これにより、プロセスネットワークモデルＤＭ１の一部の変更による影響が局所化され、プロセスネットワークモデルＤＭ１の他の部分に与える影響を小さくできる。そのため、安定性を確保しつつ、プロセスネットワークモデルＤＭ１を変更することがより容易になる。また、プロセスネットワークモデルＤＭ１が階層的に表現されることにより、プロセス同士の間関係をより容易に把握できるため、安定した業務運営がさらに容易になる。以上のことから、本実施形態によれば、可塑性と安定性とを両立させることによって変化の速度が速いビジネス環境への適応度が高いプロセス実行システム１００を提供することがより容易になる。

[0167] （第２の実施形態）

第２実施形態に係るプロセス実行システム２００は、概説すれば、プロセスネットワーク状態管理部が特定のプロセスノードの状態を変更するプロセス状態変更要求を受け付ける点、プロセスマスタノードが非同期フラグマスタおよび自動起動フラグマスタ、並びにプロセスノードが非同期フラグおよび自動起動フラグを有する点で、第１実施形態に係るプロセス実行システム１００とは異なる。第２実施形態に係るプロセス実行システム２００によれば、非同期処理を含むビジネスプロセスの実行が容易になる。

[0168] 非同期処理とは、例えば、顧客に対して何らかの問い合わせをし、問い合わせに対する返答があったときに特定の処理を実行するような処理である。顧客に対して何らかの問い合わせをしてから返信を受信するまでの間は待ち時間が発生するが、待ち時間の間も、プロセス実行システム２００は別の実行可能なプロセスノードを実行できるように構成しておくことが好ましい。第２実施形態に係るプロセス実行システム２００によれば、そうした非同期

処理を含むビジネスプロセスの実行が容易になる。なお、第2実施形態に係るプロセス実行システム200は、以下で特に説明する構成を除いて、第1実施形態に係るプロセス実行システム100と同様に構成できる。

[0169] 図18は、本実施形態に係るプロセス実行システム200と、プロセス実行システム200と通信するアクション実行システム50の構成を示す図である。図19は、プロセス実行システム200のブロック図である。図20は、プロセス実行システム200の制御部210の機能ブロック図である。図21は、プロセス実行システム200のプロセスネットワーク状態管理部260の機能ブロック図である。図22は、プロセス実行システム200のプロセスネットワークモデル入力部151が入力を受け付けるプロセスネットワークモデルDM2のデータ構造を例示する模式図である。図23は、プロセス実行システム200のプロセスネットワーク生成部250によって生成されるプロセスネットワークD2のデータ構造を例示する模式図である。図24は、プロセス実行システム200のプロセスネットワークモデル入力部151が入力を受け付けるプロセスネットワークモデルDM2のプロセスマスタノードDM20のデータ構造を例示する模式図である。図25は、プロセス実行システム200のプロセスネットワーク生成部250によって生成されるプロセスネットワークD2のプロセスノードD20のデータ構造を例示する模式図である。以下、第2実施形態に係るプロセス実行システム200について説明する。

[0170] 図18に示すように、本実施形態に係るプロセス実行システム200は、アクション実行システム50との間で通信手段90を介して情報を送受信する。アクション実行システム50は、第1実施形態において説明したアクション実行システム50と同様に構成される。通信手段90は、第1実施形態において説明した通信手段90と同様に構成される。

[0171] 図19に示すように、本実施形態に係るプロセス実行システム200は、制御部210、記憶部220および通信部230を備える。これらの構成要素は、バス240を介して互いに接続される。記憶部220は、第1実施形

態に係るプロセス実行システム１００の記憶部１２０と同様に構成される。通信部２３０は、第１実施形態に係るプロセス実行システム１００の通信部１３０と同様に構成される。バス２４０は、第１実施形態に係るプロセス実行システム１００のバス１４０と同様に構成される。

[0172] 図２０に示すように、制御部２１０は、プロセスネットワーク生成部２５０、プロセスネットワーク状態管理部２６０、プロセスネットワーク実行部２７０として機能する。プロセスネットワーク生成部２５０は、第１実施形態に係るプロセス実行システム１００のプロセスネットワーク生成部１５０と同様に構成される。プロセスネットワーク実行部２７０は、第１実施形態に係るプロセス実行システム１００のプロセスネットワーク実行部１７０と同様に構成される。

[0173] 図２１に示すように、プロセスネットワーク状態管理部２６０は、プロセス状態変更要求受付部２６１を有する。

[0174] プロセス状態変更要求受付部２６１は、特定のプロセスノードＤ２０の状態を変更するプロセス状態変更要求を受け付ける。プロセスネットワーク状態管理部２６０は、プロセス状態変更要求受付部２６１が受け付けたプロセス状態変更要求に基づいてプロセスネットワークＤ１に含まれる当該特定のプロセスノードＤ２０の状態を変更する。プロセス状態変更要求受付部２６１がプロセス状態変更要求を受け付ける対象のプロセスノードＤ２０は、プロセス状態変更要求の送信側によってランダムに選択可能である。「ランダムに選択可能」における「ランダム」とは、プロセス実行システム２００の内部状態とは因果関係を特定できないことを意味する。すなわち、「ランダムに選択可能」とは、プロセス状態変更要求受付部２６１から見たときに、プロセス状態変更要求が行われるタイミングや、プロセス状態変更要求が行われるプロセスノードＤ２０を予測できないということを意味する。プロセス状態変更要求受付部２６１がプロセス状態変更要求を受け付ける対象のプロセスノードＤ２０は、実行可能な状態に状態遷移したことの無いプロセスノードＤ２０であり得る。

- [0175] プロセス状態変更要求受付部261は、プロセス実行システム200の外部のシステムからプロセス状態変更要求を受け付け可能に構成される。プロセス状態変更要求受付部261は、プロセス状態変更要求を受け付けるAPIを有するように構成してもよいし、プロセス状態変更要求を受け付けるGUIを有するように構成してもよい。プロセス状態変更要求を受け付けるAPIは、例えば、HTTPを使用した通信によってプロセス状態変更要求を受け付けるように構成できる。また、プロセス状態変更要求を受け付けるGUIは、例えば、ブラウザからの操作によってプロセス状態変更要求を受け付けるように構成できる。
- [0176] 図22に示すように、本実施形態に係るプロセスネットワークモデルDM2は、プロセスマスタノードDM20を複数含み、プロセスマスタノードDM20同士の間の関係を表現している。好ましくは、プロセスネットワークモデルDM2は、複数のプロセスマスタノードDM20同士の間の関係を階層的に表現した階層型のプロセスネットワークモデルDM1である。プロセスネットワークモデルDM2は、以下に特に説明する点を除いて、第1実施形態に係るプロセスネットワークモデルDM1と同様に構成される。
- [0177] 図23に示すように、本実施形態に係るプロセスネットワークD2は、プロセスノードD20を複数含み、プロセスノードD20同士の間の関係を表現している。好ましくは、プロセスネットワークD2は、複数のプロセスノードD20同士の間の関係を階層的に表現した階層型のプロセスネットワークD1である。プロセスネットワークD2は、以下に特に説明する点を除いて、第1実施形態に係るプロセスネットワークD1と同様に構成される。
- [0178] 図24を参照して、プロセスマスタノードDM20は、プロセスマスタノード識別子DM21と、親プロセスマスタノード識別子DM22と、プロセス生成情報DM25と、を有する。プロセスマスタノード識別子DM21は、第1実施形態に係るプロセスマスタノード識別子DM11と同様に構成できる。親プロセスマスタノード識別子DM22は、第1実施形態に係る親プロセスマスタノード識別子DM12と同様に構成できる。

- [0179] 図25を参照して、プロセスノードD20は、プロセスノード識別子D21と、親プロセスノード識別子D22と、プロセス状態D23と、プロセス定義情報D25と、を有する。プロセスノード識別子D21は、第1実施形態に係るプロセスノード識別子D11と同様に構成できる。親プロセスノード識別子D22は、第1実施形態に係る親プロセスノード識別子D12と同様に構成できる。
- [0180] プロセス生成情報DM25は、プロセス番号マスタDM251と、プロセス名マスタDM252と、アクション情報マスタDM253と、非同期フラグマスタDM254と、自動起動フラグマスタDM255と、を含む。プロセス番号マスタDM251は、第1実施形態に係るプロセス番号マスタDM151と同様に構成できる。プロセス名マスタDM252は、第1実施形態に係るプロセス名マスタDM152と同様に構成できる。アクション情報マスタDM253は、第1実施形態に係るアクション情報マスタDM153と同様に構成できる。
- [0181] プロセス定義情報D25は、プロセス番号D251と、プロセス名D252と、アクション情報D253と、非同期フラグD254と、自動起動フラグD255と、を含む。プロセス番号D251は、第1実施形態に係るプロセス番号D151と同様に構成できる。プロセス名D252は、第1実施形態に係るプロセス名D152と同様に構成できる。アクション情報D253は、第1実施形態に係るアクション情報D153と同様に構成できる。
- [0182] 非同期フラグマスタDM254の値および非同期フラグD254の値は、例えば真理値である。プロセスネットワーク生成部250は、非同期フラグマスタDM254の値がB（真理値TrueまたはFalse）の場合に、プロセスネットワーク生成部250において生成されるプロセスノードD20の非同期フラグの値がBになるように構成できる。
- [0183] 自動起動フラグマスタDM255の値および自動起動フラグD255の値は、例えば真理値である。プロセスネットワーク生成部250は、自動起動フラグマスタDM255の値がB（真理値TrueまたはFalse）の場

合に、プロセスネットワーク生成部250において生成されるプロセスノードD20の自動起動フラグD255の値がBになるように構成できる。

[0184] 次に図26および図27を参照し、本実施形態に係るプロセス実行システム200の動作を説明する。図26は、プロセス実行システム200において実行される処理の手順を表すフローチャートである。図27は、プロセス実行システム200において実行される実行可能なプロセスノードD20の実行処理の手順を表すフローチャートである。図26および図27のフローチャートに示す処理は、プログラムに従い、プロセス実行システム200の制御部210により実行される。

[0185] 図26に示すように、プロセス実行システム200の制御部210により実行されるプログラムは、プロセスネットワークモデルDM2の入力を受け付けるステップS21と、プロセスネットワークD2を生成するステップS22と、プロセスノードD20の状態遷移処理を行うステップS23と、実行可能なプロセスノードD20の実行処理を行うステップS24と、プロセスネットワークD2に含まれる全てのプロセスノードD20が完了状態になっているか判断するステップS25と、を有する。プロセスネットワークモデルDM2の入力を受け付けるステップS21は、第1実施形態に係るプロセスネットワークモデルDM1の入力を受け付けるステップS11と同様に構成できる。プロセスネットワークD1を生成するステップS22は、第1実施形態に係るプロセスネットワークD1を生成するステップS12と同様に構成できる。プロセスネットワークD2に含まれる全てのプロセスノードD20が完了状態になっているか判断するステップS25は、第1実施形態に係るプロセスネットワークD1に含まれる全てのプロセスノードD10が完了状態になっているか判断するステップS15と同様に構成できる。

[0186] (ステップS23)

プロセスノードD20の状態遷移処理を行うステップS23は、第1実施形態に係るプロセス実行システム100のプロセスノードD10の状態遷移処理を行うステップS13と次の点を除いて同様に構成される。すなわち、

プロセスネットワーク状態管理部260は、プロセスノードD20を実行可能状態にするステップS1326（図13参照）およびステップS13245（図14参照）において、プロセスノードD20が自動起動プロセスである場合には当該プロセスノードD20のプロセス状態D23を実行可能状態にし、プロセスノードD20が自動起動プロセスでない場合には当該プロセスノードD20のプロセス状態D23は変化させない。プロセスノードD20が自動起動プロセスであるか否かの判断は、当該プロセスノードD20の自動起動フラグD255の値に基づいて判断する。例えば、プロセスネットワーク状態管理部260は、当該プロセスノードD20の自動起動フラグの値がTrueの場合には自動起動プロセスであると判断し、自動起動フラグの値がFalseの場合には自動起動プロセスではないと判断する。

[0187] （ステップS24）

図27に示すように、ステップS24において行われる実行可能なプロセスノードD20の実行処理は、実行可能なプロセスノードD20が存在するかどうか判断するステップS241と、実行可能なプロセスノードD20を実行するステップS242と、実行を完了したプロセスノードD20が非同期プロセスか否か判断するステップS243と、実行を完了したプロセスノードD20の状態を完了状態にするステップS244と、を有する。実行可能なプロセスノードD20が存在するかどうか判断するステップS241は、第1実施形態に係る実行可能なプロセスノードD10が存在するかどうか判断するステップS141と同様に構成できる。実行可能なプロセスノードD20を実行するステップS242は、第1実施形態に係る実行可能なプロセスノードD10を実行するステップS142と同様に構成できる。

[0188] （ステップS243）

プロセスネットワーク実行部270は、実行を完了したプロセスノードD20が非同期プロセスか否か判断する。実行を完了したプロセスノードD20が非同期プロセスであると判断した場合には処理を終了する。実行を完了したプロセスノードD20が非同期プロセスでないと判断した場合にはステ

ップS 2 4 4に進む。プロセスネットワーク実行部2 7 0は、実行を完了したプロセスノードD 2 0の非同期フラグの値に基づいて、実行を完了したプロセスノードD 2 0が非同期プロセスか否か判断する。具体的には、プロセスネットワーク実行部2 7 0は、実行を完了したプロセスノードD 2 0の非同期フラグの値がT r u eの場合に非同期プロセスと判断し、実行を完了したプロセスノードD 2 0の非同期フラグの値がF a l s eの場合に非同期プロセスではない判断する。プロセスネットワーク実行部2 7 0は、実行を完了したプロセスノードD 2 0が非同期プロセスである場合には、プロセス状態D 2 3を待機中状態に変更するように構成してもよい。

[0189] (ステップS 2 4 4)

プロセスネットワーク実行部2 7 0は、実行を完了したプロセスノードD 2 0のプロセス状態D 2 3を完了状態にする。プロセス状態D 2 3を完了状態にする方法は特に限定されないが、例えば、第1実施形態のステップS 1 4 3の説明で上述した記憶部2 2 0に対して要求を送信する方法であってよい。また、プロセスネットワーク実行部2 7 0は、実行を完了したプロセスノードD 2 0のプロセス状態D 2 3を完了状態にするプロセス状態変更要求をプロセスネットワーク状態管理部2 6 0に行ってもよい。

[0190] (作用効果)

本実施形態に係るプロセス実行システム2 0 0において、プロセスネットワーク状態管理部2 6 0は、特定のプロセスノードD 2 0の状態を変更するプロセス状態変更要求を受け付けるプロセス状態変更要求受付部2 6 1を備え、当該プロセス状態変更要求受付部2 6 1が受け付けたプロセス状態変更要求に基づいてプロセスネットワークD 1に含まれる当該特定のプロセスノードD 2 0の状態を変更する。

[0191] これにより、本実施形態に係るプロセス実行システム2 0 0によって実行されるプロセスネットワークモデルDM 2は、非同期処理を含むビジネスプロセスを表現し易くなる。そのため、業務内容の変更等に伴ってプロセスネットワークモデルDM 2を変更するのがより容易になるとともに、プロセス

ネットワークモデルDM2のメンテナンス性も向上する。以上のことから、本実施形態によれば、可塑性と安定性とを両立させることによって変化の速度が速いビジネス環境への適応度が高いプロセス実行システム200を提供することがさらに容易になる。

[0192] (第3の実施形態)

第3実施形態に係るプロセス実行システム300は、概説すれば、制御部310が転写部380をさらに有する点において第2実施形態に係るプロセス実行システム200とは異なる。第3実施形態に係るプロセス実行システム300によれば、環境Eの変化に応じて実行されるプロセスの内容を変化させることができるため、変化の速度が速いビジネス環境への適応度をより向上させることができる。なお、第3実施形態に係るプロセス実行システム300は、以下で特に説明する構成を除いて、第2実施形態に係るプロセス実行システム200と同様に構成できる。

[0193] 図28は、本実施形態に係るプロセス実行システム300の制御部310の機能ブロック図である。図29は、本実施形態に係るプロセス実行システム300の転写部380の機能ブロック図である。図30は、本実施形態に係るプロセス実行システム300のプロセス生成情報DM35を示す概念図である。図31は、本実施形態に係るプロセス実行システム300と環境Eとの間の相互作用を示す概念図である。図32は、本実施形態に係るプロセス実行システム300が参照するケースデータテーブルCDTのデータ構造を例示する模式図である。図33は、本実施形態に係るプロセス実行システム300のプロセスネットワークモデル入力部351が保存するコンテキストデータC×Dのデータ構造を例示する模式図である。図34は、本実施形態に係るプロセス実行システム300の転写プロセスタグPTの構文を例示する図である。図35は、本実施形態に係るプロセス実行システム300の選択プロセスタグPTsの構文を例示する図である。図36は、本実施形態に係るプロセス実行システムの反復プロセスタグPTiの構文を例示する図である。図37は、本実施形態に係るプロセス実行システム300の置換プ

ロセスタグPT rの構文を例示する図である。図38は、本実施形態に係るプロセス実行システム300の転写因子モデルタグMTの構文を例示する図である。図39は、本発明の第3実施形態に係るプロセス実行システム300の転写因子モデルMの出力データDoutを例示する図である。以下、第3実施形態に係るプロセス実行システム300について説明する。

- [0194] 図28から図31を参照して、本実施形態に係るプロセス実行システム300は、一のプロセスマスタノードDM30が有するプロセス生成情報DM35を、当該一のプロセスマスタノードDM30に対応する一のプロセスノードD30にプロセス定義情報D35として転写する転写プロセスを実行する転写部380をさらに有する。
- [0195] プロセス生成情報DM35は、転写プロセスを定義した一または複数の転写プロセス定義情報PDinfoを含む。具体的には、プロセス生成情報DM35は、当該プロセス生成情報DM35を、当該プロセス生成情報DM35を有するプロセスマスタノードDM30に対応するプロセスノードD30にプロセス定義情報D35として転写する転写プロセスを定義した一または複数の転写プロセス定義情報PDinfoを含む。転写プロセス定義情報PDinfoは、プロセス実行システム300が通信可能な環境Eが転写プロセスに与える影響をモデル化した転写因子モデルMを定義した一または複数の転写因子モデル定義情報MDinfoを含む。
- [0196] 転写部380は、転写プロセスの実行要求を受け付ける転写要求受付部381と、転写因子モデル定義情報MDinfoによって定義された転写因子モデルMを評価する転写因子モデル評価部382と、転写プロセス定義情報PDinfoによって定義された転写プロセスを実行する転写プロセス実行部383と、を有する。
- [0197] プロセス実行システム300が通信可能な環境Eは、プロセス実行システム300が排他的に通信可能な内部環境Einと、プロセス実行システム300から見た外部のシステムSysOutが通信可能な外部環境Eoutと、を含む。

- [0198] 外部のシステム *SysOut* は、アクション実行システム 50 を含む。外部のシステム *SysOut* は、アクション実行システム 50 以外のシステムであってプロセス実行システム 300 と通信するシステム *S1* を含む。外部のシステム *SysOut* は、プロセス実行システム 300 と直接的に通信しないシステム *S2* を含む。プロセス実行システム 300 と直接的に通信しないシステムとは、プロセス実行システム 300 から発信される信号を原因とする因果関係を特定できないシステムを意味する。
- [0199] 内部環境 *Ein* は、プロセス実行システム 300 の記憶部 220 を含む。内部環境 *Ein* は、データベースサーバーとして機能する記憶部 220 を含む。内部環境 *Ein* は、プロセス実行システム 300 がサーバーマシン上で動作する *Web* アプリとして構成されている場合、当該 *Web* アプリが管理するデータベースサーバーを含む。
- [0200] 外部環境 *Eout* は、プロセス実行システム 300 と通信する外部のシステム *SysOut* そのものを含む。
- [0201] 転写因子モデル評価部 382 は、プロセス実行システム 300 が通信可能な環境 *E* から取得できる情報を入力データ *Din* として転写因子モデル *M* を評価する。転写因子モデル評価部 382 が転写因子モデル *M* を評価するのに使用する入力データ *Din* は、プロセス実行システム 300 が通信できる環境 *E* から取得できる情報である限りにおいて特に限定されない。
- [0202] 入力データ *Din* は、プロセス実行システム 300 の記憶部 220（内部環境 *Ein*）から取得したデータを含む。
- [0203] 入力データ *Din* は、外部のシステム *SysOut* が通信する外部データベースサーバー（外部環境 *Eout*）から取得したデータを含む。
- [0204] 外部のシステム *SysOut* が通信する外部データベースサーバー（外部環境 *Eout*）からのデータの取得は、当該外部のシステム *SysOut* を介して行ってもよい。例えば、外部のシステム *SysOut* が *Django* などの *Web* アプリ作成用のフレームワークを使用して構築された *Web* アプリの場合、当該 *Web* アプリが通信する外部データベース（外部環境 *Eout*）

ut)からのデータ取得は、当該WebアプリのAPIを介して行うことができる。また、外部のシステムSysOutがMicrosoft 365のPower Automate、外部データベースサーバー（外部環境Eout）がMicrosoft Dataverse、SharePoint OnlineまたはExcel Onlineなどの場合、外部データベースサーバーからのデータの取得は、Power Automateのフローで構成したAPI（HTTPのPOSTで起動する）を介して行うことができる。

[0205] また、外部のシステムSysOutが通信する外部データベースサーバー（外部環境Eout）からのデータの取得は、当該外部データベースサーバーから直接的に行ってもよい。例えば、外部のシステムSysOutがMicrosoft 365のPower Automate、外部データベースサーバー（外部環境Eout）がMicrosoft Dataverseの場合、Microsoft Dataverseからのデータの取得は、Microsoft Dataverse Web APIを介して行うことができる。

[0206] 外部のシステムSysOutは、Ethereumなどのブロックチェーンを使用したWeb3.0（Web3）関連システムを含む。外部のシステムSysOutは、ブロックチェーン上に構成されたDAppやスマートコントラクトを処理するアプリを含む。

[0207] 入力データDinは、プロセス実行システム300が通信可能な外部のシステムSysOut（外部環境Eout）が生成した情報であってもよい。プロセス実行システム300が通信可能な外部のシステムSysOut（外部環境Eout）が生成した情報は、例えば、プロセス実行システム300からの要求に応じて、学習済の生成モデルによって生成されたテキストや画像、音声などのデータを含む。クライアントからの要求に応じて、学習済の生成モデルを使用してテキストを生成するサービスとして、例えば、ChatGPTなどがある。

- [0208] プロセスネットワークモデル入力部351は、第2実施形態に係るプロセス実行システム200のプロセスネットワークモデル入力部251と次の点を除いて同様に構成される。すなわち、プロセスネットワークモデル入力部351は、プロセスネットワークモデルDM3の入力を受け付ける際に、当該プロセスネットワークモデルDM3から生成される一のプロセスネットワークD3に関連付けられるコンテキストデータC×Dの入力を受け付ける。入力データD_{in}は、プロセスネットワークモデル入力部351が入力を受け付けたコンテキストデータC×Dを含む。
- [0209] プロセスネットワークモデル入力部351は、プロセスネットワークモデルDM3の入力を受け付ける際に入力を受け付けたコンテキストデータC×Dを、記憶部220として機能するデータベースサーバーに保存する。
- [0210] 図32を参照して、プロセスネットワークモデルDM3が表現する業務において処理するケースに関するケースデータCDは、記憶部220として機能するデータベースサーバー（内部環境E_{in}）の一または複数のケースデータテーブルCDTにケース毎に保存されている。ケースデータテーブルCDTは、外部のシステムS_{ys}O_{ut}が通信する外部データベースサーバー（外部環境E_{out}）に保存されていてもよい。
- [0211] 図33を参照して、プロセスネットワークモデル入力部351は、プロセスネットワークモデルDM3の入力を受け付ける際に入力を受け付けたコンテキストデータC×Dを、記憶部220として機能するデータベースサーバーのコンテキストテーブルC×Tに保存する。コンテキストデータC×Dは、プロセスネットワークモデルDM3が表現する業務において処理するケースのケースID（C_ID）を含む。ケースID（C_ID）は、ケースデータテーブルCDTを管理するアプリケーション名C_ID₁、ケースデータテーブルCDTのテーブル名C_ID₂、およびケースデータテーブルCDTにおける該当ケースのケースデータが保存されているレコードのレコードID（C_ID₃）の組み合わせで表現される。
- [0212] プロセスネットワークモデル入力部351は、記憶部220として機能す

るデータベースサーバーにコンテキストデータ $C \times D$ を保存する際に、当該コンテキストデータ $C \times D$ が関連付けられているプロセスネットワーク $D3$ に含まれるプロセスノード $D30$ のプロセスノード識別子 $D31$ と対応付けて保存する。

[0213] 転写因子モデル M は、入力データ D_{in} に応じて出力結果が定まるモデルであれば特に限定されない。転写因子モデル M は、決定論的モデルおよび確率論的モデルのいずれであってもよい。転写因子モデル M として使用される決定論的モデルは、関数によって表現されたモデル、好ましくは関数を引数にする高階関数によって表現されたモデルを含む。転写因子モデル M として使用される決定論的モデルは、決定論的な深層学習モデルなどの機械学習モデルを含む。転写因子モデル M として使用される確率論的モデルは、確率論的な深層学習モデル（生成モデル）やディープボルツマンマシンなどの機械学習モデルを含む。

[0214] 転写プロセス実行部 383 は、転写因子モデル評価部 382 が評価した一または複数の転写因子モデル M の評価結果 R に基づいて転写プロセスを制御する。

[0215] 転写プロセス実行部 383 が制御する転写プロセスは、転写因子モデル M の評価結果 R を条件にしてプロセス生成情報 $DM35$ の一部を選択する選択プロセス、転写因子モデル M の評価結果 R を条件にしてプロセス生成情報 $DM35$ の一部を反復する反復プロセス、および転写因子モデル M の評価結果 R によってプロセス生成情報 $DM35$ の一部を置換する置換プロセスを含む。

[0216] 転写プロセス実行部 383 は、選択プロセス、反復プロセスおよび置換プロセス以外にも、Web アプリ作成用のフレームワークが備えているテンプレートエンジンと同等の機能を実現するように構成できる。例えば、転写プロセス実行部 383 は、Web アプリ作成用のフレームワークである Django で使用されている Django Template Language (DTL) が提供するテンプレートタグやフィルタなどを用いた機能と同

等の機能を実現するように構成できる。

- [0217] 転写プロセス実行部383は、Webアプリ作成用のフレームワークが備えているテンプレートエンジンを拡張して構成できる。具体的には、転写プロセス実行部383は、プロセス生成情報DM35をテンプレートとしてテンプレートエンジンに入力するとともに、転写因子モデルMの評価結果Rをコンテキストデータとして入力することによって、置換プロセス、選択プロセスおよび反復プロセスを含む様々な操作をテンプレートエンジンに実行させることができる。転写プロセス実行部383は、プロセス生成情報DM35をテンプレートとしてテンプレートエンジンに入力する際に、プロセス生成情報DM35に含まれる転写プロセスタグPTおよび転写因子モデルタグMTを、テンプレートエンジンの仕様で定められている構文に基づいて書き換える。
- [0218] 転写プロセス定義情報PDinfoは、所定の構文で記述された転写プロセスタグPTによって表現される。転写プロセスタグPTを記述する構文は特に限定されない。図34から図37に、転写プロセスタグPTの構文の定義を拡張バックス・ナウア記法(Extended Backus-Naur Form)によって例示する。
- [0219] 図34を参照して、転写プロセスタグPTは、選択プロセスを表現する選択プロセスタグPTs、反復プロセスを表現する反復プロセスタグPTiまたは置換プロセスを表現する置換プロセスタグPTRである。
- [0220] 図35を参照して、選択プロセスタグPTsは、＜主選択ブロック＞と、選択的に＜副選択ブロック＞を繰り返し並べたデータと、選択的に＜デフォルト選択ブロック＞と、＜選択プロセスタグ終端記号＞と、を順に並べたデータである。
- [0221] ＜主選択ブロック＞は、＜主選択ブロック開始記号＞と、＜選択条件＞と、＜転写プロセスタグ終端記号＞と、＜選択ブロック＞と、を順に並べたデータである。
- [0222] ＜主選択ブロック開始記号＞は、主選択ブロックの開始位置を表現する。

＜主選択ブロック開始記号＞は、＜転写プロセスタグ開始記号＞と「i f」
とを並べたものを例示しているが、主選択ブロックの開始位置を特定できる
限りにおいて特に限定されない。＜転写プロセスタグ開始記号＞は、転写プ
ロセスタグP Tの開始位置を表現する。＜転写プロセスタグ開始記号＞は「
{%」である場合を例示しているが、転写プロセスタグP Tの開始位置を特
定できる限りにおいて特に限定されない。

[0223] ＜選択条件＞は、＜論理式＞と、選択的に＜論理演算子＞および＜論理式＞
を順に並べたデータと、を順に並べたデータである。＜論理式＞は、＜式＞
と、＜論理演算子＞と、＜式＞と、を順に並べたデータである。＜式＞は、
＜値＞、＜転写因子モデルタグ＞、または＜論理式＞を括弧で括ったデー
タである。＜式＞に＜転写因子モデルタグ＞を指定することによって、＜
転写因子モデルタグ＞で表現される転写因子モデルMの評価結果Rを＜選択
条件＞に用いることができる。また、＜式＞に＜論理式＞を指定すること
によって、＜論理式＞の中で＜論理式＞を使うことができる。＜論理式＞の
評価は、最も内側の括弧で括られている論理式の評価が優先される。＜値＞
は、文字列型のデータである。

[0224] ＜転写プロセスタグ終端記号＞は、転写プロセスタグP Tの終端位置を表
現する。＜転写プロセスタグ終端記号＞は「}%」である場合を例示してい
るが、転写プロセスタグP Tの終端位置を特定できる限りにおいて特に限定
されない。

[0225] ＜副選択ブロック＞は、＜副選択ブロック開始記号＞と、＜選択条件＞と、
＜転写プロセスタグ終端記号＞と、＜選択ブロック＞と、を順に並べたデ
ータである。＜副選択ブロック開始記号＞は、副選択ブロックの開始位置を
表現する。＜副選択ブロック開始記号＞は、＜転写プロセスタグ開始記号＞
と「e l i f」とを並べたものを例示しているが、副選択ブロックの開始位
置を特定できる限りにおいて特に限定されない。

[0226] ＜デフォルト選択ブロック＞は、＜デフォルト選択ブロック開始記号＞と、
＜転写プロセスタグ終端記号＞と、＜選択ブロック＞と、を順に並べたデ

ータである。＜デフォルト選択ブロック開始記号＞は、デフォルト選択ブロックの開始位置を表現する。＜デフォルト選択ブロック開始記号＞は、＜転写プロセスタグ開始記号＞と「e l s e」とを並べたものを例示しているが、デフォルト選択ブロックの開始位置を特定できる限りにおいて特に限定されない。

[0227] ＜選択ブロック＞は、選択的に＜文字列＞と、選択的に＜転写プロセスタグ＞とを並べたデータをゼロ回以上並べたデータである。例えば、＜選択ブロック＞は、＜文字列＞、＜文字列＞、＜転写プロセスタグ＞、＜文字列＞、＜転写プロセスタグ＞を順に並べたデータであり得る。＜選択ブロック＞は、空であってもよい。

[0228] ＜選択プロセスタグ終端記号＞は、選択プロセスタグ P T s の終端位置を表現する。＜選択プロセスタグ終端記号＞は、＜転写プロセスタグ開始記号＞と「e n d i f」と＜転写プロセスタグ終端記号＞を並べたものを例示しているが、選択プロセスタグ P T s の終端位置を特定できる限りにおいて特に限定されない。

[0229] 図 3 6 を参照して、反復プロセスタグ P T i は、＜反復条件ブロック＞と、＜反復ブロック＞と、＜反復プロセスタグ終端記号＞と、を順に並べたデータである。

[0230] ＜反復条件ブロック＞は、＜反復条件ブロック開始記号＞と、＜反復条件＞と、＜転写プロセスタグ終端記号＞と、を順に並べたデータである。

[0231] ＜反復条件ブロック開始記号＞は、反復条件ブロックの開始位置を表現する。＜反復条件ブロック開始記号＞は、＜転写プロセスタグ開始記号＞と「f o r」とを並べたものを例示しているが反復条件ブロックの開始位置を特定できる限りにおいて特に限定されない。

[0232] ＜反復条件＞は、＜ループ変数＞と、「i n」と、＜ループ変数配列＞と、を順に並べたデータである。＜ループ変数＞は文字列型のデータである。＜ループ変数配列＞は、＜ループ変数配列定数＞または＜転写因子モデルタグ＞である。＜ループ変数配列＞として＜転写因子モデルタグ＞を指定可能

であることによって、転写因子モデルMの評価結果Rをループ変数配列として使用できる。＜ループ変数配列定数＞は、文字列型のデータを「,」区切りで並べたものを括弧で括ったデータである。

[0233] ＜反復ブロック＞は、選択的に＜文字列＞と、選択的に＜ループ変数＞を二重括弧（「{ {」、} }」）で括ったデータと、選択的に＜転写プロセスタグ＞と、を順に並べたデータをゼロ回以上並べたデータである。例えば、＜反復ブロック＞は、{ {＜ループ変数＞} }、＜文字列＞、＜文字列＞、＜転写プロセスタグ＞、{ {＜ループ変数＞} }、＜文字列＞を順に並べたデータであり得る。＜反復ブロック＞は空であってもよい。

[0234] ＜反復プロセスタグ終端記号＞は、反復プロセスタグP T iの終端位置を表現する。＜反復プロセスタグ終端記号＞は、＜転写プロセスタグ開始記号＞と「e n d f o r」と＜転写プロセスタグ終端記号＞を順に並べたものを例示しているが、反復プロセスタグP T iの終端位置を特定できる限りにおいて特に限定されない。

[0235] 図37を参照して、置換プロセスタグP T rは、＜転写因子モデルタグ＞そのものである。

[0236] 転写因子モデル定義情報M D i n f oは、所定の構文で記述された転写因子モデルタグM Tによって表現される。転写因子モデルタグM Tを記述する構文は特に限定されない。図38に、転写因子モデルタグM Tの構文の定義を拡張バックス・ナウア記法（E x t e n d e d B a c k u s - N a u r F o r m）によって例示する。

[0237] 転写因子モデルタグM Tは、＜転写因子モデルタグ開始記号＞と、＜転写因子モデル＞と、選択的に＜評価結果セクタ群＞と、＜転写因子モデルタグ終端記号＞と、を順に並べたデータである。

[0238] ＜転写因子モデルタグ開始記号＞は、転写因子モデルタグM Tの開始位置を表現する。＜転写因子モデル＞は、転写因子モデルMを表現する。＜評価結果セクタ群＞は、転写因子モデルMの出力データD o u tの中から転写因子モデルMの評価結果Rとして使用するデータを表現する。＜転写因子モ

デルタグ終端記号>は、転写因子モデルタグMTの終端位置を表現する。

[0239] 図38では、<転写因子モデルタグ開始記号>が「<<<」である場合を例示しているが、<転写因子モデルタグ開始記号>は転写因子モデルタグMTの開始位置を特定できる限りにおいて特に限定されない。

[0240] <転写因子モデル>は、<モデルID>と、選択的に<入力データ群>と、を順に並べて構成される。<モデルID>は、転写因子モデルMを一意に識別するための識別子である。<入力データ群>は、転写因子モデルMを評価するのに使用する一または複数の入力データD_{in}を表現する。

[0241] <モデルID>の表現方法は特に限定されない。本実施形態において、<モデルID>は階層構造を有する。具体的には、<モデルID>は、一または複数の<モデルグループ名>を「.」で区切って並べたデータと、<モデル名>と、を順に並べたデータである。<モデルグループ名>は、左側から右側に向かって上位の階層から下位の階層になるように並べられる。<モデルグループ名>および<モデル名>は、文字列型のデータである。

[0242] <入力データ群>は、<入力データ>を括弧で括ったデータである。<入力データ>は、一または複数の<入力変数>を「,」で区切って並べたデータである。

[0243] <入力変数>は、<入力変数名>と<入力変数の値>を「=」で連結したデータである。<入力変数名>は文字列型のデータである。<入力変数の値>は、<文字列>または<転写因子モデルタグ>である。転写因子モデルMは、<入力変数の値>に<転写因子モデルタグ>を設定可能であることによって、一の転写因子モデルMの評価結果Rを他の転写因子モデルMの入力データD_{in}として取る高階関数の構造を有する。

[0244] <評価結果セクタ群>は、<評価結果セクタ>を「.」で区切って並べたデータである。<評価結果セクタ>は、転写因子モデルMの出力データD_{out}の中から転写因子モデルMの評価結果Rとして使用するデータを識別する識別子を表現する。転写因子モデルMの出力データD_{out}に含まれるデータ全てを転写因子モデルMの評価結果Rとして使用する場合には、

＜評価結果セクタ群＞はなくてもよい。

[0245] 転写因子モデルMの出力データD o u t は、複数のデータを含み得る。具体的には、転写因子モデルMの出力データD o u t は、リスト、配列、連想配列、およびそれらが入れ子になった構造化されたデータであり得る。図39は、転写因子モデルMの出力データD o u t がJ S O N形式である場合を例示している。転写因子モデルMの出力データD o u t が構造化されたデータである場合、＜評価結果セクタ群＞には、当該構造化されたデータの上位の階層から下位の階層へと左側から右側に向かって各階層におけるデータを特定する識別子が＜評価結果セクタ＞として順に並べられる。例えば、図39に例示する出力データD o u t において「値3 1 B」を指定する＜評価結果セクタ群＞は、「1. キー1 2. 0. キー2 2. キー3 1」であり得る。

[0246] 転写因子モデルMが確率論的モデルの場合、出力データD o u t は確率分布であり得る。出力データD o u t が確率分布である場合、＜評価結果セクタ群＞は、例えば、当該確率分布において最大確率をとる確率変数の値を表現するデータであり得る。

[0247] 図38では、＜転写因子モデルタグ終端記号＞が「>>>」である場合を例示しているが、＜転写因子モデルタグ終端記号＞は転写因子モデルタグM T の終端位置を特定できる限りにおいて特に限定されない。

[0248] 次に、図40および図41を参照し、プロセス実行システム300の動作を説明する。図40は、本実施形態に係るプロセス実行システム300において実行される処理の手順を表すフローチャートである。図41は、本実施形態に係るプロセス実行システム300において実行される実行可能なプロセスノードD30の実行処理の手順を表すフローチャートである。図41および図42のフローチャートに示す処理は、プログラムに従い、プロセス実行システム300の制御部310により実行される。

[0249] 図40に示すように、プロセス実行システム300の制御部310により実行されるプログラムは、プロセスネットワークモデルDM3の入力を受け

付けるステップS 3 1 と、プロセスネットワークD 3 を生成するステップS 3 2 と、プロセスノードD 3 0 の状態遷移処理を行うステップS 3 3 と、実行可能なプロセスノードD 3 0 の実行処理を行うステップS 3 4 と、プロセスネットワークD 3 に含まれる全てのプロセスノードD 3 0 が完了状態になっているか判断するステップS 3 5 と、を有する。

[0250] プロセスネットワークD 3 を生成するステップS 3 2 は、第2実施形態に係るプロセスネットワークD 2 を生成するステップS 2 2 と同様に構成できる。プロセスノードD 3 0 の状態遷移処理を行うステップS 3 3 は、第2実施形態に係るプロセスノードD 2 0 の状態遷移処理を行うステップS 2 3 と同様に構成できる。プロセスネットワークD 3 に含まれる全てのプロセスノードD 3 0 が完了状態になっているか判断するステップS 3 5 は、第2実施形態に係るプロセスネットワークD 2 に含まれる全てのプロセスノードD 2 0 が完了状態になっているか判断するステップS 2 5 と同様に構成できる。

[0251] (ステップS 3 1)

プロセスネットワークモデルDM 3 の入力を受け付けるステップS 3 1 は、第2実施形態に係るプロセス実行システム2 0 0 のプロセスネットワークモデルDM 2 の入力を受け付けるステップS 2 1 と次の点を除いて同様に構成される。すなわち、プロセスネットワークモデル入力部3 5 1 は、プロセスネットワークモデルDM 3 の入力を受け付ける際に、プロセスネットワークモデルDM 3 から生成される一のプロセスネットワークD 3 に関連付けられるコンテキストデータC × D の入力を受け付ける。プロセスネットワークモデル入力部3 5 1 は、プロセスネットワークモデルDM 3 の入力を受け付ける際に入力を受け付けたコンテキストデータC × D を、記憶部2 2 0 として機能するデータベースサーバーのコンテキストテーブルC × T に保存する。

[0252] (ステップS 3 4)

図4 1 に示すように、ステップS 3 4 において行われる実行可能なプロセスノードD 3 0 の実行処理は、実行可能なプロセスノードD 3 0 が存在する

か否か判断するステップS 3 4 1と、転写プロセスの実行要求を送信するステップS 3 4 2と、実行可能なプロセスノードD 3 0を実行するステップS 3 4 3と、実行を完了したプロセスノードD 3 0が非同期プロセスか否か判断するステップS 3 4 4と、実行を完了したプロセスノードD 3 0の状態を完了状態にするステップS 3 4 5と、を有する。

[0253] 実行可能なプロセスノードD 3 0が存在するか否か判断するステップS 3 4 1は、第2実施形態に係る実行可能なプロセスノードD 2 0が存在するか否か判断するステップS 2 4 1と同様に構成できる。実行可能なプロセスノードD 3 0を実行するステップS 3 4 3は、第2実施形態に係る実行可能なプロセスノードD 2 0を実行するステップS 2 4 2と同様に構成できる。実行を完了したプロセスノードD 3 0が非同期プロセスか否か判断するステップS 3 4 4は、第2実施形態に係る実行を完了したプロセスノードD 2 0が非同期プロセスか否か判断するステップS 2 4 3と同様に構成できる。実行を完了したプロセスノードD 3 0の状態を完了状態にするステップS 3 4 5は、第2実施形態に係る実行を完了したプロセスノードD 2 0の状態を完了状態にするステップS 2 4 4と同様に構成できる。

[0254] (ステップS 3 4 2)

プロセスネットワーク実行部3 7 0は、実行可能なプロセスノードD 3 0に対応するプロセスマスタノードDM 3 0が有するプロセス生成情報DM 3 5を当該実行可能なプロセスノードD 3 0にプロセス定義情報D 3 5として転写する転写プロセスの実行要求を送信する。転写部3 8 0は、転写要求受付部3 8 1が転写プロセスの実行要求を受け付けたプロセス生成情報DM 3 5を、当該プロセス生成情報DM 3 5を有するプロセスマスタノードDM 3 0に対応するプロセスノードにプロセス定義情報D 3 5として転写する。プロセス定義情報D 3 5は、転写部3 8 0が転写することによって更新される。

[0255] (ステップS 3 4 3)

プロセスネットワーク実行部3 7 0は、ステップS 3 4 2において転写部

380が転写することによって更新されたプロセス定義情報D35に基づいてプロセスを実行する。

[0256] 図42から図44を参照し、プロセス実行システム300の転写部380により実行されるプログラムの動作を説明する。図42は、本実施形態に係るプロセス実行システム300の転写部380において実行される処理の手順を表すフローチャートである。図43は、本実施形態に係るプロセス実行システム300の転写部380において実行される評価対象転写因子モデルMeの評価処理の手順を表すフローチャートである。図44は、本実施形態に係るプロセス実行システム300の転写部380において実行される原評価対象転写因子モデルMoの評価処理の手順を表すフローチャートである。図42から図44のフローチャートに示す処理は、プログラムに従い、プロセス実行システム300の制御部310により実行される。

[0257] プロセス実行システム300の転写部380により実行されるプログラムは、転写プロセスの実行要求を受け付けるステップS381と、プロセス生成情報DM35を転写情報Tinfoに設定するステップS382と、転写情報Tinfoから転写プロセス定義情報PDinfoを抽出するステップS383と、転写プロセス定義情報PDinfoが抽出されたか否か判断するステップS384と、評価対象転写因子モデルMeを設定するステップS385と、評価対象転写因子モデルMeの評価処理を行うステップS386と、転写情報Tinfoを加工するステップS387と、全ての転写プロセス定義情報PDinfoが処理されたか否か判断するステップS388と、転写プロセス定義情報PDinfoによって定義された転写プロセスを実行するステップS389と、を有する。

[0258] (ステップS381)

転写要求受付部381は、一のプロセスマスタノードDM30が有するプロセス生成情報DM35を転写する転写プロセスの実行要求を受け付ける。

[0259] (ステップS382)

転写プロセス実行部383は、ステップS381において転写プロセスの

実行要求を受け付けたプロセス生成情報DM35を転写情報Tinfoに設定する。

[0260] (ステップS383)

転写プロセス実行部383は、転写情報Tinfoから転写プロセス定義情報PDinfoを抽出する。具体的には、転写プロセス実行部383は、転写情報Tinfoから転写プロセス定義情報PDinfoを表現する転写プロセスタグPTを抽出する。

[0261] (ステップS384)

転写プロセス実行部383は、ステップS383において転写プロセス定義情報PDinfoが抽出されたか否か判断する。具体的には、転写プロセス実行部383は、ステップS383において転写プロセス定義情報PDinfoを表現する転写プロセスタグPTが抽出されたか否か判断する。転写プロセス定義情報PDinfoが抽出されたと判断したらステップS385に進む。転写プロセス定義情報PDinfoが抽出されなかったと判断したらステップS389に進む。

[0262] (ステップS385)

転写因子モデル評価部382は、転写プロセス定義情報PDinfoに含まれる転写因子モデル定義情報MDinfoによって定義された転写因子モデルMを評価対象転写因子モデルMeに設定する。

[0263] 具体的には、転写因子モデル評価部382は、転写プロセス定義情報PDinfoを表現する転写プロセスタグPTが選択プロセスタグPTsである場合、選択プロセスタグPTsの選択条件に含まれている転写因子モデルタグMTによって表現された転写因子モデルMを評価対象転写因子モデルMeに設定する。転写因子モデル評価部382は、転写プロセス定義情報PDinfoを表現する転写プロセスタグPTが反復プロセスタグPTiである場合、反復プロセスタグPTiの反復条件に含まれている転写因子モデルタグMTによって表現された転写因子モデルMを評価対象転写因子モデルMeに設定する。転写因子モデル評価部382は、転写プロセス定義情報PDinfo

f oを表現する転写プロセスタグP Tが置換プロセスタグP T rである場合、置換プロセスタグP T rをなす転写因子モデルタグM Tによって表現された転写因子モデルMを評価対象転写因子モデルM eに設定する。

[0264] (ステップS 3 8 6)

転写因子モデル評価部3 8 2は、ステップS 3 8 5において設定された評価対象転写因子モデルM eの評価処理を行う。転写因子モデル評価部3 8 2は、評価対象転写因子モデルM eの評価処理を行う際に、プロセス実行システム3 0 0が通信可能な環境Eから取得できる情報を入力データD i nとして評価対象転写因子モデルM eを評価する。

[0265] (ステップS 3 8 7)

転写プロセス実行部3 8 3は、ステップS 3 8 6において評価した転写因子モデルMの評価結果Rに基づいて転写情報T i n f oを加工する。

[0266] 転写プロセス実行部3 8 3は、転写プロセス定義情報P D i n f oを表現する転写プロセスタグP Tが選択プロセスタグP T sである場合、選択プロセスタグP T sの選択条件に含まれる転写因子モデルタグM Tが表現する転写因子モデルMの評価結果Rに基づいて選択条件を評価し、選択条件が真になる選択ブロックによって選択プロセスタグP T sを置換する。

[0267] 転写プロセス実行部3 8 3は、転写プロセス定義情報P D i n f oを表現する転写プロセスタグP Tが反復プロセスタグP T iである場合、＜ループ変数配列＞に含まれる要素の各々をループ変数の値として反復ブロックを反復して生成した反復情報によって反復プロセスタグP T iを置換する。転写プロセス実行部3 8 3は、反復情報を生成する際に、反復ブロック内に「 { <ループ変数> } 」が含まれる場合にはループ変数の値によって「 { { <ループ変数> } } 」を置換する。転写プロセス実行部3 8 3は、＜ループ変数配列＞に転写因子モデルタグM Tが設定されている場合には、当該転写因子モデルタグM Tが表現する転写因子モデルMの評価結果Rを＜ループ変数配列＞として扱う。

[0268] 転写プロセス実行部3 8 3は、転写プロセス定義情報P D i n f oを表現

する転写プロセスタグPTが置換プロセスタグPT_rである場合、置換プロセスタグPT_rをなす転写因子モデルタグMTが表現する転写因子モデルMの評価結果Rによって置換プロセスタグPT_rを置換する。

[0269] (ステップS388)

転写プロセス実行部383は、ステップS383において抽出された転写プロセス定義情報PDinfoが全て処理されたか判断する。転写プロセス定義情報PDinfoが全て処理されたと判断したらステップS383に戻る。処理されていない転写プロセス定義情報PDinfoが存在する場合には、ステップS385に戻る。

[0270] (ステップS389)

転写プロセス実行部383は、ステップS381において転写プロセスの実行要求を受け付けたプロセスマスタノードDM30に対応するプロセスノードD30に転写情報Tinfoをプロセス定義情報D35として転写する。

[0271] 図43を参照して、評価対象転写因子モデルMeの評価処理を行うステップS386は、評価対象転写因子モデルMeを原評価対象転写因子モデルMoに設定するステップS3861と、入力データDinに設定されている転写因子モデルMを抽出するステップS3862と、転写因子モデルMが抽出されたか否か判断するステップS3863と、抽出された転写因子モデルMを評価対象転写因子モデルMeに設定するステップS3864と、評価対象転写因子モデルMeの評価処理を行うステップS3865と、抽出された転写因子モデルMを全て評価したか否か判断するステップS3866と、原評価対象転写因子モデルMoの評価処理を行うステップS3867と、を有する。

[0272] (ステップS3861)

転写因子モデル評価部382は、評価対象転写因子モデルMeを原評価対象転写因子モデルMoに設定する。

[0273] (ステップS3862)

転写因子モデル評価部 382 は、原評価対象転写因子モデル M_o の入力データ D_{in} に設定されている転写因子モデル M を抽出する。具体的には、転写因子モデル評価部 382 は、原評価対象転写因子モデル M_o を表現する転写因子モデルタグ MT の＜入力変数の値＞に設定されている転写因子モデルタグ MT が表現する転写因子モデル M を抽出する。

[0274] (ステップ S 3863)

転写因子モデル評価部 382 は、ステップ S 3862 において転写因子モデル M が抽出されたか否かを判断する。転写因子モデル M が抽出されたと判断した場合には、ステップ S 3864 に進む。転写因子モデル M が抽出されていないと判断した場合には、ステップ S 3867 に進む。

[0275] (ステップ S 3864)

転写因子モデル評価部 382 は、ステップ S 3862 において抽出された転写因子モデル M を評価対象転写因子モデル M_e に設定する。

[0276] (ステップ S 3865)

転写因子モデル評価部 382 は、評価対象転写因子モデル M_e の評価処理を行う。

[0277] (ステップ S 3866)

転写因子モデル評価部 382 は、ステップ S 3862 において抽出された転写因子モデル M が全て評価されたか否かを判断する。ステップ S 3862 において抽出された転写因子モデル M が全て評価されたと判断した場合には、ステップ S 3867 に進む。ステップ S 3862 において抽出された転写因子モデル M のうち、評価されていない転写因子モデル M があると判断した場合には、ステップ S 3864 に戻る。

[0278] (ステップ S 3867)

転写因子モデル評価部 382 は、原評価対象転写因子モデル M_o の評価処理を行う。

[0279] 図 44 を参照して、原評価対象転写因子モデル M_o の評価処理を行うステップ S 3867 は、原評価対象転写因子モデル M_o のモデルタイプを特定す

るステップS 3 8 6 7 1 と、原評価対象転写因子モデルM o の出力データD o u t を取得するステップS 3 8 6 7 2 と、原評価対象転写因子モデルM o の評価結果Rを決定するステップS 3 8 6 7 3 と、を有する。

[0280] (ステップS 3 8 6 7 1)

転写因子モデル評価部3 8 2 は、原評価対象転写因子モデルM o のモデルタイプを特定する。具体的には、転写因子モデル評価部3 8 2 は、原評価対象転写因子モデルM o を表現する転写因子モデルタグMTに基づいて原評価対象転写因子モデルM o のモデルタイプを特定する。より具体的には、転写因子モデル評価部3 8 2 は、原評価対象転写因子モデルM o を表現する転写因子モデルタグMTに含まれる<モデルID>に基づいて原評価対象転写因子モデルM o のモデルタイプを特定する。

[0281] さらに具体的には、転写因子モデル評価部3 8 2 は、原評価対象転写因子モデルM o を表現する転写因子モデルタグMTに含まれる<モデルID>において、<モデルグループ名>がなく、かつ、モデル名が「自プロセス」の場合には、原評価対象転写因子モデルM o のモデルタイプを自プロセス参照タイプと特定する。

[0282] 転写因子モデル評価部3 8 2 は、原評価対象転写因子モデルM o を表現する転写因子モデルタグMTに含まれる<モデルID>において、<モデルグループ名>がアプリ名、かつ、<モデル名>がテーブル名の場合には、原評価対象転写因子モデルM o のモデルタイプをテーブル参照タイプと特定する。

[0283] 転写因子モデル評価部3 8 2 は、原評価対象転写因子モデルM o を表現する転写因子モデルタグMTに含まれる<モデルID>において、<モデルグループ名>がなく、かつ、<モデル名>が「スクリプト」の場合には、原評価対象転写因子モデルM o のモデルタイプをスクリプトタイプと特定する。

[0284] 転写因子モデル評価部3 8 2 は、原評価対象転写因子モデルM o を表現する転写因子モデルタグMTに含まれる<モデルID>において、<モデルグループ名>がなく、かつ、<モデル名>が「機械学習モデル」の場合には、

原評価対象転写因子モデルM_oのモデルタイプを機械学習モデルタイプと特定する。

[0285] 転写因子モデル評価部382は、原評価対象転写因子モデルM_oを表現する転写因子モデルタグに含まれる<モデルID>において、<モデルグループ名>がなく、かつ、モデル名が「API」の場合には、原評価対象転写因子モデルM_oのモデルタイプをAPIタイプと特定する。

[0286] (ステップS38672)

転写因子モデル評価部382は、プロセス実行システム300が通信可能な環境Eから取得できる情報を入力データD_{in}として原評価対象転写因子モデルM_oを評価して出力データD_{out}を取得する。具体的には、転写因子モデル評価部382は、ステップS38671において特定した原評価対象転写因子モデルM_oのモデルタイプに基づいて原評価対象転写因子モデルM_oを評価して出力データD_{out}を取得する。

[0287] より具体的には、転写因子モデル評価部382は、ステップS38671において特定した原評価対象転写因子モデルM_oのモデルタイプが自プロセス参照タイプの場合には、自プロセスノードD30の情報を出力データD_{out}として取得する。

[0288] 転写因子モデル評価部382は、ステップS38671において特定した原評価対象転写因子モデルM_oのモデルタイプがテーブル参照タイプの場合には、原評価対象転写因子モデルM_oを表現する転写因子モデルタグMTの<モデルグループ名>（テーブルを管理するアプリ名を表現）および<モデル名>（テーブル名を表現）によって特定されるテーブルから、当該転写因子モデルタグMTの入力データD_{in}に基づいて検索した情報を出力データD_{out}として取得する。原評価対象転写因子モデルM_oの出力データD_{out}として情報を取得するテーブルは、内部環境E_{in}または外部環境E_{out}のいずれにあってもよい。

[0289] 転写因子モデル評価部382は、ステップS38671において特定した原評価対象転写因子モデルM_oのモデルタイプがスクリプトタイプの場合に

は、原評価対象転写因子モデルM_oを表現する転写因子モデルタグMTの入力データD_{in}に基づいて実行したスクリプトの実行結果を出力データD_{out}として取得する。

[0290] 転写因子モデル評価部382は、ステップS38671において特定した原評価対象転写因子モデルM_oのモデルタイプが機械学習モデルタイプの場合には、原評価対象転写因子モデルM_oを表現する転写因子モデルタグMTの入力データD_{in}に基づいて実行した機械学習モデルを用いた推論結果を出力データD_{out}として取得する。

[0291] 転写因子モデル評価部382は、ステップS38671において特定した原評価対象転写因子モデルM_oのモデルタイプがAPIタイプの場合には、原評価対象転写因子モデルM_oを表現する転写因子モデルタグMTの入力データD_{in}に基づいて実行したAPIの実行結果を出力データD_{out}として取得する。

[0292] (ステップS38673)

転写因子モデル評価部382は、ステップS38672において取得した原評価対象転写因子モデルM_oの出力データD_{out}から、原評価対象転写因子モデルM_oの評価結果Rとして返却するデータを決定する。具体的には、転写因子モデル評価部382は、原評価対象転写因子モデルM_oを表現する転写因子モデルタグMTに基づいて原評価対象転写因子モデルM_oの出力データD_{out}から評価結果Rとして返却するデータを決定する。より具体的には、転写因子モデル評価部382は、原評価対象転写因子モデルM_oを表現する転写因子モデルタグMTの<評価結果セクタ群>に基づいて原評価対象転写因子モデルM_oの出力データD_{out}から評価結果Rとして返却するデータを決定する。

[0293] 図45から図47を参照して、図42を用いて説明した転写部380において実行される転写プロセスの実行処理の動作例を説明する。図45は、本実施形態に係るプロセス実行システム300の転写部380の動作例について説明するために使用するプロセス生成情報DM35を例示する模式図であ

る。図46は、本実施形態に係るプロセス実行システム300の転写部380の動作例について説明するために使用する転写情報Tinfoを例示する模式図である。図47は、本実施形態に係るプロセス実行システム300の転写部380の動作例について説明するために使用する転写情報Tinfoを例示する模式図である。図45に示すように、動作例の説明で使用するプロセス生成情報DM35は転写プロセス定義情報PDinfoを表現する複数の転写プロセスタグPT（PTs1、PTs2、PTs3、PTiおよびPTr）を含む。

- [0294] 転写プロセス実行部383は、転写プロセスの実行要求を転写要求受付部381が受け付けると（ステップS381）、プロセス生成情報DM35を転写情報Tinfoに設定する（ステップS382）。
- [0295] ステップS383において、転写プロセス実行部383は、転写情報Tinfoから選択プロセスタグPTs1および置換プロセスタグPTrを抽出する。ステップS383において転写プロセスタグPT（PTs1およびPTr）が抽出されるので、ステップS384の判断はYesになる。
- [0296] ステップS385からステップS387の処理は、選択プロセスタグPTs1および置換プロセスタグPTrの各々に対して行われる。ここでは、選択プロセスタグPTs1、置換プロセスタグPTrの順に処理されることとする。
- [0297] ステップS385において、転写因子モデル評価部382は、選択プロセスタグPTs1の選択条件に含まれる<転写因子モデルタグ11>が表現する転写因子モデルMを評価対象転写因子モデルMeに設定する。ステップS386において、転写因子モデル評価部382は、評価対象転写因子モデルMeの評価処理を行う。ここでは、評価対象転写因子モデルMeの評価結果Rが「値m」であるとする。
- [0298] ステップS387において、転写プロセス実行部383は、ステップS386における転写因子モデルMの評価結果Rに応じて転写情報Tinfoを加工する。具体的には、転写プロセス実行部383は、真になる選択条件「

＜転写因子モデルタグ11＞ == “値m”」に対応する選択ブロック（反復プロセスタグPTiが含まれるブロック）によって転写情報Tinfoに含まれる選択プロセスタグPTs1を置換する。

[0299] 置換プロセスタグPTrの処理が完了していないので、ステップS388における判断はNoとなり、ステップS385に戻る。

[0300] ステップS385において、転写因子モデル評価部382は、置換プロセスタグPTrをなす＜転写因子モデルタグ01＞が表現する転写因子モデルMを評価対象転写因子モデルMeに設定する。ステップS386において、転写因子モデル評価部382は、評価対象転写因子モデルMeの評価処理を行う。ここでは、評価対象転写因子モデルMeの評価結果Rが「評価結果01」であるとする。

[0301] ステップS387において、転写プロセス実行部383は、ステップS386における評価対象転写因子モデルMeの評価結果Rによって転写情報Tinfoに含まれる置換プロセスタグPTrを置換する。

[0302] ステップS383において抽出された転写プロセスタグPT（PTs1およびPTr）が処理されたので、ステップS388における判断はYesとなり、ステップS383に戻る。ここまでの処理によって、転写情報Tinfoは図46のようになる。

[0303] ステップS383において、転写プロセス実行部383は、転写情報Tinfoから反復プロセスタグPTiを抽出する。反復プロセスタグPTiが抽出されたので、ステップS384の判断はYesになる。

[0304] ステップS385において、転写因子モデル評価部382は、反復プロセスタグPTiの反復条件に含まれる＜転写因子モデルタグ22＞が表現する転写因子モデルMを評価対象転写因子モデルMeに設定する。ステップS386において、転写因子モデル評価部382は、評価対象転写因子モデルMeの評価処理を行う。ここでは、評価対象転写因子モデルMeの評価結果Rが「評価結果221」、「評価結果222」、および「評価結果223の」3つのデータを含むとする。

- [0305] ステップS 3 8 7において、転写プロセス実行部3 8 3は、ステップS 3 8 6における評価対象転写因子モデルM eの評価結果Rに含まれる3つのデータの各々をループ変数の値として反復プロセスタグP T iに含まれる<反復ブロック>を反復して反復情報を生成する。反復情報に含まれる「{ {ループ変数} }」は、評価対象転写因子モデルM eの評価結果Rの3つのデータの各々によって置換される。転写プロセス実行部3 8 3は、生成した反復情報によって転写情報T i n f oに含まれる反復プロセスタグP T iを置換する。
- [0306] ステップS 3 8 3において抽出された反復プロセスタグP T iが処理されたので、ステップS 3 8 8における判断はY e sとなり、ステップS 3 8 3に戻る。ここまでの処理によって、転写情報T i n f oは図4 7のようになる。
- [0307] ステップS 3 8 3において転写情報T i n f oから抽出される転写プロセス定義情報P D i n f oは存在しないため、ステップS 3 8 4の判断はN oとなり、ステップS 3 8 9に進む。
- [0308] ステップS 3 8 9において、転写プロセス実行部3 8 3は、ステップS 3 8 1において転写プロセスの実行要求を受け付けたプロセスマスタノードD M 3 0に対応するプロセスノードD 3 0に転写情報T i n f oをプロセス定義情報D 3 5として転写する。以上で、転写部3 8 0による転写プロセスの実行処理が終了する。
- [0309] 図4 8から図5 3を参照して、図4 3および図4 4を用いて説明した転写部3 8 0において実行される評価対象転写因子モデルM eの評価処理の動作例を説明する。図4 8は、本実施形態に係るプロセス実行システム3 0 0の転写部3 8 0における評価対象転写因子モデルM eの評価処理の動作例について説明するために使用する転写プロセスタグP Tを例示する模式図である。図4 9は、本実施形態に係るプロセス実行システム3 0 0の転写部3 8 0における評価対象転写因子モデルM eの評価処理の動作例について説明するために使用する転写プロセスタグP Tを例示する模式図である。図5 0は、

本実施形態に係るプロセス実行システム300の転写部380における評価対象転写因子モデルMeの評価処理の動作例について説明するために使用する転写プロセスタグPTを例示する模式図である。図51は、本実施形態に係るプロセス実行システム300の転写部380における評価対象転写因子モデルMeの評価処理の動作例について説明するために使用する転写プロセスタグPTを例示する模式図である。図52は、本実施形態に係るプロセス実行システム300の転写部380における評価対象転写因子モデルMeの評価処理の動作例について説明するために使用する転写プロセスタグPTを例示する模式図である。図53は、本実施形態に係るプロセス実行システム300の転写部380における評価対象転写因子モデルMeの評価処理の動作例について説明するために使用する転写プロセスタグPTを例示する模式図である。

- [0310] まず、評価対象転写因子モデルMeが図48に例示する転写因子モデルタグMTが表現する転写因子モデルMである場合について説明する。
- [0311] 図43を参照して、ステップS3861では、評価対象転写因子モデルMeを原評価対象転写因子モデルMoに設定する。ステップS3862では、入力データDinに設定されている転写因子モデルMが抽出されないため、ステップS3863の判断はNoになり、ステップS3867に進む。ステップS3867では、原評価対象転写因子モデルMoの評価処理が行われる。
- [0312] 図44を参照して、ステップS38671では、原評価対象転写因子モデルMoを表現する転写因子モデルタグMTの<モデルID>が「自プロセス」であるため、原評価対象転写因子モデルMoのモデルタイプを自プロセス参照タイプと特定する。ステップS38672では、原評価対象転写因子モデルMoのモデルタイプが自プロセス参照タイプであるため、自プロセスノードD30の情報を記憶部220から出力データDoutとして取得する。ステップS38673では、原評価対象転写因子モデルMoを表現する転写因子モデルタグMTの<評価結果セクタ群>が「プロセスノード識別子」

であるため、出力データDoutのうちプロセスノード識別子D31のデータが原評価対象転写因子モデルMoの評価結果Rとして決定される。

[0313] 次に、評価対象転写因子モデルMeが図49に例示する転写因子モデルタグMTが表現する転写因子モデルMである場合について説明する。

[0314] 図43を参照して、ステップS3861では、評価対象転写因子モデルMeを原評価対象転写因子モデルMoに設定する。ステップS3862では、＜入力変数名＞「プロセスノード識別子」に対応する＜入力変数の値＞に設定されている転写因子モデルMを抽出する。ステップS3862において転写因子モデルMが抽出されるため、ステップS3863の判断はYesになり、ステップS3864に進む。ステップS3864において、＜入力変数名＞「プロセスノード識別子」に対応する＜入力変数の値＞に設定されている転写因子モデルMが評価対象転写因子モデルMeに設定される。ステップS3865では、評価対象転写因子モデルMeの評価処理が行われる。評価対象転写因子モデルMeは、図48に例示する転写因子モデルタグMTが表現する転写因子モデルMと同じであるため、図48に例示する転写因子モデルタグMTについて上述した評価処理と同様の処理が行われる。ステップS3862において抽出された転写因子モデルMが全て評価されたため、ステップS3866の判断はYesになり、ステップS3867に進む。ステップS3867では、原評価対象転写因子モデルMoの評価処理が行われる。

[0315] 図44を参照して、ステップS38671では、原評価対象転写因子モデルMoを表現する転写因子モデルタグMTの＜モデルID＞が「アプリVa、コンテキストテーブル」であるため、原評価対象転写因子モデルMoのモデルタイプをテーブル参照タイプと特定する。ステップS38672では、原評価対象転写因子モデルMoのモデルタイプがテーブル参照タイプであるため、アプリVaが管理するコンテキストテーブルから、入力データDinに基づいて検索した情報を出力データDoutとして取得する。ここでは、アプリVaが管理するコンテキストテーブルから、プロセスノード識別子D31が自プロセスノードD30のプロセスノード識別子D31であるレコー

ドの情報が出力データDoutとして取得される。ステップS38673では、原評価対象転写因子モデルMoを表現する転写因子モデルタグMTの<評価結果セレクトタ群>が「ケースID」であるため、出力データDoutのうちケースIDのデータが原評価対象転写因子モデルMoの評価結果Rとして決定される。

[0316] 次に、評価対象転写因子モデルMeが図50に例示する転写因子モデルタグMTが表現する転写因子モデルMである場合について説明する。

[0317] 図43を参照して、ステップS3861では、評価対象転写因子モデルMeを原評価対象転写因子モデルMoに設定する。ステップS3862では、<入力変数名>「列Cp1」に対応する<入力変数の値>に設定されている転写因子モデルMを抽出する。ステップS3862において転写因子モデルMが抽出されるため、ステップS3863の判断はYesになり、ステップS3864に進む。ステップS3864において、<入力変数名>「列Cp1」に対応する<入力変数の値>に設定されている転写因子モデルMが評価対象転写因子モデルMeに設定される。ステップS3865では、評価対象転写因子モデルMeの評価処理が行われる。評価対象転写因子モデルMeは、図49に例示する転写因子モデルタグMTが表現する転写因子モデルMと同じであるため、図49に例示する転写因子モデルタグMTについて上述した評価処理と同様の処理が行われる。ステップS3862において抽出された転写因子モデルMが全て評価されたため、ステップS3866の判断はYesになり、ステップS3867に進む。ステップS3867では、原評価対象転写因子モデルMoの評価処理が行われる。

[0318] 図44を参照して、ステップS38671では、原評価対象転写因子モデルMoを表現する転写因子モデルタグMTの<モデルID>が「アプリWa、テーブルWt」であるため、原評価対象転写因子モデルMoのモデルタイプをテーブル参照タイプと特定する。ステップS38672では、原評価対象転写因子モデルMoのモデルタイプがテーブル参照タイプであるため、アプリWaが管理するテーブルWtから、入力データDinに基づいて検索し

た情報を出力データD o u tとして取得する。ここでは、アプリW aが管理するテーブルW tから、列C p 1が自プロセスノードD 3 0に関連付けられたコンテキストデータC x DのケースI D、かつ、列C p 2が「列C p 2値」であるレコードの情報が出力データD o u tとして取得される。ステップS 3 8 6 7 3では、原評価対象転写因子モデルM oを表現する転写因子モデルタグM Tの<評価結果セクタ群>が「列W s」であるため、出力データD o u tのうち列W sのデータが原評価対象転写因子モデルM oの評価結果Rとして決定される。

[0319] 次に、評価対象転写因子モデルM eが図5 1に例示する転写因子モデルタグM Tが表現する転写因子モデルMである場合について説明する。

[0320] 図4 3を参照して、ステップS 3 8 6 1では、評価対象転写因子モデルM eを原評価対象転写因子モデルM oに設定する。ステップS 3 8 6 2では、<入力変数名>「スクリプトコード」に対応する<入力変数の値>に設定されている転写因子モデルMおよび<入力変数名>「入力変数D p 1」に対応する<入力変数の値>に設定されている転写因子モデルMを抽出する。ステップS 3 8 6 2において転写因子モデルMが抽出されるため、ステップS 3 8 6 3の判断はY e sになり、ステップS 3 8 6 4に進む。ステップS 3 8 6 4およびステップS 3 8 6 5を経て、ステップS 3 8 6 2において抽出された転写因子モデルMの評価処理が行われる。

[0321] <入力変数名>「スクリプトコード」に対応する<入力変数の値>に設定されている転写因子モデルMはテーブル参照タイプであるため、その評価処理は図4 8および図4 9に例示する転写因子モデルタグM Tについて上述した評価処理と同様である。具体的には、アプリX aが管理するスクリプトテーブルから、スクリプトI Dが「スクリプトI D値」であるレコードを出力データD o u tとして取得し、当該出力データD o u tからスクリプトコードのデータが評価結果Rとして決定される。

[0322] <入力変数名>「入力変数D p 1」に対応する<入力変数の値>に設定されている転写因子モデルMは、図4 9に例示する転写因子モデルタグM Tが

表現する転写因子モデルMと同じであるため、図49に例示する転写因子モデルタグMTについて上述した評価処理と同様の処理が行われる。

[0323] ステップS3862において抽出された転写因子モデルMが全て評価されたため、ステップS3866の判断はYesになり、ステップS3867に進む。ステップS3867では、原評価対象転写因子モデルMoの評価処理が行われる。

[0324] 図44を参照して、ステップS38671では、原評価対象転写因子モデルMoを表現する転写因子モデルタグMTの<モデルID>が「スクリプト」であるため、原評価対象転写因子モデルMoのモデルタイプをスクリプトタイプと特定する。ステップS38672では、原評価対象転写因子モデルMoのモデルタイプがスクリプトタイプであるため、入力データDinに基づいて実行したスクリプトの実行結果を出力データDoutとして取得する。ここでは、<入力変数名>「スクリプトコード」に対応する<入力変数の値>に設定されている転写因子モデルMの評価結果Rをスクリプトコードとし、スクリプトタイプを「Python」としてスクリプトを実行する。スクリプトを実行する際に、スクリプトの入力変数Dp1の値は、<入力変数名>「入力変数Dp1」に対応する<入力変数の値>に設定されている転写因子モデルMの評価結果Rとし、入力変数Dp2の値は「入力変数Dp2値」とする。ステップS38673では、<評価結果セクタ群>「評価結果セクタDs1、評価結果セクタDs2」に基づいて出力データDoutから評価結果Rとして返却するデータが決定される。

[0325] 次に、評価対象転写因子モデルMeが図52に例示する転写因子モデルタグが表現する転写因子モデルである場合について説明する。

[0326] 図43を参照して、ステップS3861では、評価対象転写因子モデルMeを原評価対象転写因子モデルMoに設定する。ステップS3862では、<入力変数名>「機械学習モデルデータ」に対応する<入力変数の値>に設定されている転写因子モデルMおよび<入力変数名>「入力変数Ep1」に対応する<入力変数の値>に設定されている転写因子モデルMを抽出する。

ステップS 3 8 6 2において転写因子モデルMが抽出されるため、ステップS 3 8 6 3の判断はY e sになり、ステップS 3 8 6 4に進む。ステップS 3 8 6 4およびステップS 3 8 6 5を経て、ステップS 3 8 6 2において抽出された転写因子モデルMの評価処理が行われる。

[0327] <入力変数名>「機械学習モデルデータ」に対応する<入力変数の値>に設定されている転写因子モデルMはテーブル参照タイプであるため、その評価処理は図4 8および図4 9に例示する転写因子モデルタグMTについて上述した評価処理と同様である。具体的には、アプリY aが管理する機械学習モデルテーブルから、機械学習モデルI Dが「機械学習モデルI D値」であるレコードを出力データD o u tとして取得し、当該出力データD o u tからモデルデータが評価結果Rとして決定される。

[0328] <入力変数名>「入力変数E p 1」に対応する<入力変数の値>に設定されている転写因子モデルMは、図4 9に例示する転写因子モデルタグMTが表現する転写因子モデルMと同じであるため、図4 9に例示する転写因子モデルタグMTについて上述した評価処理と同様の処理が行われる。

[0329] ステップS 3 8 6 2において抽出された転写因子モデルMが全て評価されたため、ステップS 3 8 6 6の判断はY e sになり、ステップS 3 8 6 7に進む。ステップS 3 8 6 7では、原評価対象転写因子モデルM oの評価処理が行われる。

[0330] 図4 4を参照して、ステップS 3 8 6 7 1では、原評価対象転写因子モデルM oを表現する転写因子モデルタグMTの<モデルI D>が「機械学習モデル」であるため、原評価対象転写因子モデルM oのモデルタイプを機械学習モデルタイプと特定する。ステップS 3 8 6 7 2では、原評価対象転写因子モデルM oのモデルタイプが機械学習モデルタイプであるため、入力データD i nに基づいて実行した機械学習モデルの推論結果を出力データD o u tとして取得する。ここでは、<入力変数名>「機械学習モデルデータ」に対応する<入力変数の値>に設定されている転写因子モデルMの評価結果Rを機械学習モデルデータとし、モデルフレームワークを「P y T o r c h」

として機械学習モデルを用いた推論処理を実行する。機械学習モデルを用いた推論処理を実行する際に、機械学習モデルの入力変数E p 1の値は、＜入力変数名＞「入力変数E p 1」に対応する＜入力変数の値＞に設定されている転写因子モデルMの評価結果Rとし、機械学習モデルの入力変数E p 2の値は「入力変数E p 2値」とする。ステップS 3 8 6 7 3では、＜評価結果セクタ群＞「評価結果セクタE s 1」に基づいて出力データD o u tから評価結果Rとして返却するデータが決定される。

[0331] 次に、評価対象転写因子モデルM eが図5 3に例示する転写因子モデルタグMTが表現する転写因子モデルMである場合について説明する。

[0332] 図4 3を参照して、ステップS 3 8 6 1では、評価対象転写因子モデルM eを原評価対象転写因子モデルM oに設定する。ステップS 3 8 6 2では、＜入力変数名＞「A P Iアドレス」に対応する＜入力変数の値＞に設定されている転写因子モデルMおよび＜入力変数名＞「キーG p 1」に対応する＜入力変数の値＞に設定されている転写因子モデルMを抽出する。ステップS 3 8 6 2において転写因子モデルMが抽出されるため、ステップS 3 8 6 3の判断はY e sになり、ステップS 3 8 6 4に進む。ステップS 3 8 6 4およびステップS 3 8 6 5を経て、ステップS 3 8 6 2において抽出された転写因子モデルMの評価処理が行われる。

[0333] ＜入力変数名＞「A P Iアドレス」に対応する＜入力変数の値＞に設定されている転写因子モデルMはテーブル参照タイプであるため、その評価処理は図4 8および図4 9に例示する転写因子モデルタグMTについて上述した評価処理と同様である。具体的には、アプリZ aが管理するA P Iテーブルから、A P I I Dが「A P I I D値」であるレコードを出力データD o u tとして取得し、当該出力データD o u tからA P Iアドレスが評価結果Rとして決定される。

[0334] ＜入力変数名＞「キーG p 1」に対応する＜入力変数の値＞に設定されている転写因子モデルMは、図4 9に例示する転写因子モデルタグMTが表現する転写因子モデルMと同じであるため、図4 9に例示する転写因子モデル

タグMTについて上述した評価処理と同様の処理が行われる。

- [0335] ステップS 3 8 6 2において抽出された転写因子モデルMが全て評価されたため、ステップS 3 8 6 6の判断はY e sになり、ステップS 3 8 6 7に進む。ステップS 3 8 6 7では、原評価対象転写因子モデルM oの評価処理が行われる。
- [0336] 図4 4を参照して、ステップS 3 8 6 7 1では、原評価対象転写因子モデルM oを表現する転写因子モデルタグMTの<モデルID>が「API」であるため、原評価対象転写因子モデルM oのモデルタイプをAPIタイプと特定する。ステップS 3 8 6 7 2では、原評価対象転写因子モデルM oのモデルタイプがAPIタイプであるため、入力データD i nに基づいて実行したAPIの実行結果を出力データD o u tとして取得する。ここでは、<入力変数名>「APIアドレス」に対応する<入力変数の値>に設定されている転写因子モデルMの評価結果RをAPIアドレスとしてAPIを実行する。APIを実行する際に、キーG p 1の値は、<入力変数名>「キーG p 1」に対応する<入力変数の値>に設定されている転写因子モデルMの評価結果Rとし、キーG p 2の値は「キーG p 2値」とする。ステップS 3 8 6 7 3では、<評価結果セクタ群>「評価結果セクタG s 1, 評価結果セクタG s 2, 評価結果セクタG s 3」に基づいて出力データD o u tから評価結果Rとして返却するデータが決定される。
- [0337] 図5 4から図5 7を参照して、転写プロセス定義情報P D i n f oを含むプロセス生成情報D M 3 5に対する転写部3 8 0の動作例を説明する。図5 4は、本実施形態に係るプロセス実行システム3 0 0の転写部3 8 0の動作例について説明するために使用するアクション情報マスタD M 3 5 3を例示する模式図である。図5 5は、本実施形態に係るプロセス実行システム3 0 0の転写部3 8 0の動作例について説明するために使用するアクション情報D 3 5 3を例示する模式図である。図5 6は、本実施形態に係るプロセス実行システム3 0 0の転写部3 8 0の動作例について説明するために使用するアクション情報マスタD M 3 5 3を例示する模式図である。図5 7は、本実

施形態に係るプロセス実行システム300の転写部380の動作例について説明するために使用するアクション情報D353を例示する模式図である。

[0338] 図54は選択プロセスタグPTsを含むアクション情報マスタDM353を例示している。図54のアクション情報マスタDM353は、置換プロセスタグPTRをなす<転写因子モデルタグB>、<転写因子モデルタグC>、および<転写因子モデルタグD>をさらに含む。ここでは、図54に例示する選択プロセスタグPTsの選択条件に含まれる<転写因子モデルタグA>の評価結果Rが「値1」ではないとする。また、<転写因子モデルタグB>の評価結果Rは<評価結果B>であり、<転写因子モデルタグC>の評価結果Rは<評価結果C>であり、<転写因子モデルタグD>の評価結果Rは<評価結果D>であるとする。この場合、アクション情報マスタDM353を転写したアクション情報D353は、図55のようになる。

[0339] 図56は、反復プロセスタグPTiを含むアクション情報マスタDM353を例示している。図56のアクション情報マスタDM353は、置換プロセスタグPTRをなす<転写因子モデルタグB>をさらに含む。ここでは、反復プロセスタグPTiの反復条件に含まれる<転写因子モデルタグA>の評価結果Rは<評価結果A1>と<評価結果A2>の2つのデータを含むものとする。また、<転写因子モデルタグB>の評価結果Rは<評価結果B>であるとする。この場合、アクション情報マスタDM353を転写したアクション情報D353は、図57のようになる。

[0340] 図54から図57を用いて上述した説明では、転写プロセス定義情報PDinfoが、プロセス生成情報DM35のうちアクション情報マスタDM353に含まれる場合を例に説明したが、転写プロセス定義情報PDinfoがアクション情報マスタDM353以外のプロセス生成情報DM35に含まれている場合についても本動作例によって理解できる。例えば、プロセス生成情報DM35のうち、転写プロセス定義情報PDinfoがプロセス番号マスタDM351に含まれていてもよい。具体的には、プロセス番号マスタDM351は、転写因子モデル定義情報MDinfoによって定義される転写因

子モデルMの評価結果Rに応じてプロセス番号D351が定まるように構成された転写プロセス定義情報PDinfoを含んでいてもよい。これにより、プロセス実行システム300が通信可能な環境Eに応じてプロセス番号D351を変化させることができる。そのため、プロセスネットワーク状態管理部360の状態遷移処理が図12から図14を参照して上述した状態遷移処理のようにプロセス番号D351に応じて定まる場合、プロセス実行システム300が通信可能な環境Eに応じて、実行されるプロセスノードD30の順番を変化させることができる。

[0341] (作用効果)

本実施形態に係るプロセス実行システム300は、一のプロセスマスタノードDM30が有するプロセス生成情報DM35を、当該一のプロセスマスタノードDM30に対応する一のプロセスノードにプロセス定義情報D35として転写する転写プロセスを実行する転写部380をさらに有する。プロセス生成情報DM35は、転写プロセスを定義した一または複数の転写プロセス定義情報PDinfoを含む。転写プロセス定義情報PDinfoは、プロセス実行システム300が通信可能な環境Eが転写プロセスに与える影響をモデル化した転写因子モデルMを定義した一または複数の転写因子モデル定義情報MDinfoを含む。転写部380は、転写プロセスの実行要求を受け付ける転写要求受付部381と、転写因子モデル定義情報MDinfoによって定義された転写因子モデルMを評価する転写因子モデル評価部382と、転写プロセス定義情報PDinfoによって定義された転写プロセスを実行する転写プロセス実行部383と、を備える。転写プロセス実行部383は、転写因子モデル評価部382が評価した一または複数の転写因子モデルMの評価結果Rに基づいて転写プロセスを制御する。

[0342] 本実施形態に係るプロセス実行システム300によれば、プロセス実行システム300が通信可能な環境Eが転写プロセスに与える影響をモデル化した転写因子モデルMによって、プロセスの内容を定義したプロセス定義情報D35を変化させることができる。すなわち、プロセス実行システム300

が通信可能な環境Eに応じて、プロセスネットワーク実行部370が実行するプロセスの内容を変化させることができる。そのため、本実施形態に係るプロセス実行システム300によれば、可塑性と安定性とを両立させることによって変化の速度が速いビジネス環境への適応度がより高いプロセス実行システム300を提供できる。

[0343] 本実施形態に係るプロセス実行システム300のプロセスネットワーク実行部370は、プロセス定義情報D35によって定義されたプロセスを実行する際に、転写プロセスの実行要求を転写要求受付部381に送信することによって、当該プロセス定義情報D35を生成するために使用されたプロセス生成情報DM35を転写部380が転写することによって更新されたプロセス定義情報D35に基づいてプロセスを実行する。

[0344] これにより、プロセス定義情報D35によって定義されたプロセスをプロセスネットワーク実行部370が実行する際に、プロセス実行システム300が通信可能な環境Eに応じて自動的に更新されたプロセス定義情報D35に基づいてプロセスを実行できる。そのため、本実施形態に係るプロセス実行システム300によれば、可塑性と安定性とを両立させることによって変化の速度が速いビジネス環境への適応度がさらに高いプロセス実行システム300を提供できる。

[0345] 本実施形態に係るプロセス実行システム300の転写因子モデル評価部382は、一の転写因子モデルMを評価する際に、他の転写因子モデルMの評価結果Rを当該一の転写因子モデルMの入力データD_inとして使用する。

[0346] これにより、転写因子モデル評価部382は、関数を引数に取る高階関数のような構造を有する転写因子モデルMを評価できる。そのため、プロセス実行システム300が通信可能な環境Eが転写プロセスに与える複雑な影響をモデル化できる。従って、本実施形態に係るプロセス実行システム300によれば、可塑性と安定性とを両立させることによって変化の速度が速いビジネス環境への適応度がさらに高いプロセス実行システム300を提供できる。

[0347] (改変例)

上述した実施形態3の構成を改変すると、プロセス実行システム300が通信可能な環境Eに応じてプロセスネットワークD3の構造を変化させることが可能である。

[0348] 図58は、本改変例に係るプロセス実行システム300のプロセスネットワーク生成部350によって生成されるプロセスネットワークD3のデータ構造を例示する模式図である。図59は、本改変例に係るプロセス実行システム300のプロセスネットワークモデル入力部351が入力を受け付けるプロセスネットワークモデルDM3のデータ構造を例示する模式図である。図60は、本改変例に係るプロセス実行システム300のプロセスネットワークモデル入力部351が入力を受け付けるプロセスネットワークモデルDM3のプロセスマスタノードDM30のデータ構造を例示する模式図である。図61は、本改変例に係るプロセス実行システム300のプロセスネットワーク生成部350によって生成されるプロセスネットワークD3のプロセスノードD30のデータ構造を例示する模式図である。以下、本改変例に係るプロセス実行システム300について説明する。

[0349] 図58に示すように、本改変例に係るプロセス実行システム300のプロセス定義情報D35は、当該プロセス定義情報D35を有するプロセスノードD30が活性であるか非活性であるかを表現する活性フラグD356を含む。

[0350] 図59に示すように、本改変例に係るプロセス実行システム300のプロセス生成情報DM35は、活性フラグD356を生成するために使用される活性フラグマスタDM356を含む。

[0351] 図60および図61を参照して、活性フラグマスタDM356は、転写因子モデル定義情報MDinfoによって定義される転写因子モデルMの評価結果Rに応じて活性フラグD356が定まるように構成された転写プロセス定義情報PDinfoを含む。

[0352] プロセスネットワーク状態管理部360は、プロセスノードD30が有す

るプロセス定義情報D 3 5に含まれる活性フラグD 3 5 6が非活性であることを表現している場合には、当該プロセスノードD 3 0を存在しないものとして扱う。具体的には、プロセスネットワーク状態管理部3 6 0は、プロセスノードD 3 0を抽出する際（例えば、図1 3のステップS 1 3 2 1など）に、プロセスノードD 3 0が有するプロセス定義情報D 3 5に含まれる活性フラグD 3 5 6が非活性であることを表現している場合には、当該プロセスノードD 3 0は抽出結果に含まれないようにする。これにより、プロセスノードD 3 0が有するプロセス定義情報D 3 5に含まれる活性フラグD 3 5 6が非活性であることを表現している場合には、当該プロセスノードD 3 0は実行可能状態になることはなく、実質的に存在しないものとして扱われるようになる。そのため、プロセスネットワークD 3の生成時に、プロセス実行システム3 0 0が通信可能な環境Eに応じてプロセスネットワークD 3の構造が変化することになる。

[0353] プロセスネットワーク生成部3 5 0は、プロセスネットワークD 3を生成する際に、転写プロセスの実行要求を転写要求受付部3 8 1に送信することによって、プロセスネットワークモデル入力部3 5 1が入力を受け付けたプロセスネットワークモデルDM 3に含まれる複数のプロセスマスタノードDM 3 0のうちの一のプロセスマスタノードDM 3 0が有するプロセス生成情報DM 3 5を、当該一のプロセスマスタノードDM 3 0に対応するプロセスノードD 3 0にプロセス定義情報D 3 5として転写する。

[0354] （作用効果）

本改変例に係るプロセス実行システム3 0 0のプロセス定義情報D 3 5は、当該プロセス定義情報D 3 5を有するプロセスが活性であるか非活性であるかを表現する活性フラグD 3 5 6を含む。本改変例に係るプロセス実行システム3 0 0のプロセス生成情報DM 3 5は、活性フラグD 3 5 6を生成するために使用される活性フラグマスタDM 3 5 6を含む。活性フラグマスタDM 3 5 6は、転写因子モデルMの評価結果Rに応じて活性フラグD 3 5 6が定まるように構成された転写プロセス定義情報T D i n f oを含む。プロ

セスネットワーク状態管理部360は、プロセスノードD30が有するプロセス定義情報D35に含まれる活性フラグD356が非活性であることを表現している場合には、当該プロセスノードD30を存在しないものとして扱う。

[0355] これにより、プロセス実行システム300が通信可能な環境Eに応じてプロセスネットワークD3の構造を変化させることができる。そのため、本改変例に係るプロセス実行システム300によれば、可塑性と安定性とを両立させることによって変化の速度が速いビジネス環境への適応度がさらに高いプロセス実行システム300を提供できる。

[0356] また、本改変例に係るプロセス実行システム300のプロセスネットワーク生成部350は、プロセスネットワークD3を生成する際に、転写プロセスの実行要求を転写要求受付部381に送信することによって、プロセスネットワークモデル入力部351が入力を受け付けたプロセスネットワークモデルDM3に含まれる複数のプロセスマスタノードDM30のうちの一のプロセスマスタノードDM30が有するプロセス生成情報DM35を、当該一のプロセスマスタノードDM30に対応するプロセスノードD30にプロセス定義情報D35として転写する。

[0357] これにより、プロセスネットワークD3の生成時に、プロセス実行システム300が通信可能な環境Eに応じてプロセスネットワークD3の構造を変化させることができる。そのため、本改変例に係るプロセス実行システム300によれば、可塑性と安定性とを両立させることによって変化の速度が速いビジネス環境への適応度がさらに高いプロセス実行システム300を提供できる。

[0358] (第4の実施形態)

上述した実施形態1、実施形態2および実施形態3に係るプロセス実行システムのプロセスネットワークモデル入力部が入力を受け付けるプロセスネットワークモデル(Executable Process Network)は、ビジネスプロセスを表現するために一般に使用されているBPMN

(Business Process Modeling Notation)とは異なる種々の顕著な特徴を有する。

[0359] 例えば、BPMNでは、アクティビティとアクティビティとの間に、制御フロー（AND分割、XOR分割など）を表現するゲートが配置される。一方で、上述した実施形態1、実施形態2および実施形態3に係るプロセス実行システムのプロセスネットワークモデル入力部が入力を受け付けるプロセスネットワークモデルでは、プロセスとプロセスとの間に制御フローを表現するゲートを配置する必要がない。

[0360] これは、プロセスとプロセスとの間の相互作用によって複雑な制御フローを実現できるためである。例えば、一のプロセスノードまたは他のプロセスノードのいずれか一方を、それらよりも前に実行されるプロセスノード（以下、条件分岐プロセスノード）の実行結果に応じて選択的に実行したい場合（XOR分割に相当する）には、次のようにすればよい。すなわち、条件分岐プロセスノードにおいて定義されたアクションの実行によって、一のプロセスノードまたは他のプロセスノードのいずれか一方の状態を完了状態に変更するプロセス状態変更要求をプロセス状態変更要求受付部に送信する。一のプロセスノードまたは他のプロセスノードのどちらを完了状態にするかは、条件分岐プロセスノードにおいて定義されたアクションの実行結果に応じて条件付けしておく。これにより、一のプロセスノードまたは他のプロセスノードのうち、条件分岐プロセスノードにおいて定義されたアクションの実行結果に応じて、完了状態に変更されなかったプロセスノードが実行される。なお、複雑な制御フローを必要としない逐次的なプロセスノードの実行の場合には、プロセス状態変更要求受付部は設けなくてもよい。

[0361] 近年、Networks as Things（モノとしてのネットワーク）からNetworks as Processes（プロセスとしてのネットワーク）へと視点を転換し、プロセスとプロセスとの間の相互作用を中心に世界を捉えることによって、物理系や、量子コンピューティング、知能システム、生物システム、情報システム、社会システムなどのあらゆるシ

システムを統合的に扱える可能性が示唆されている (Bob Coecke, Aleks Kissinger, “Picturing Quantum Processes: A First Course in Quantum Theory and Diagrammatic Reasoning”, Cambridge University Press, 2017. John C. Baez, Mike Stay, “Physics, Topology, Logic and Computation: A Rosetta Stone”, arXiv, 2009. Uri Alon, “An Introduction to Systems Biology: Design Principles of Biological Circuits”, 2nd edition, Chapman and Hall/CRC, 2019. Saunders Mac Lane, “Categories for the Working Mathematician”, Springer, 1978. Ludwig Von Bertalanffy, “General System Theory”, Allen Lane, 1972.)。プロセス同士の相互作用を表現するネットワークに着目し、実行可能なプロセスネットワーク (Executable Process Network) を用いた本発明の実施形態に係るプロセス実行システムは、Networks as Processesへの視点の転換によって、他のあらゆるシステムとの相互接続を容易にし得る。そのため、本発明の実施形態に係るプロセス実行システムは、可塑性と安定性とを両立させることによって変化の速度が速いビジネス環境への適応度を容易に高め得る。

[0362] 以下に説明する実施形態4および実施形態5は、プロセスネットワークモデルにおいてプロセスとプロセスとの間に制御フローを表現するゲートを配置する必要がないという顕著な特徴によって有利な効果を発揮する構成を有する。

- [0363] 第4実施形態に係るプロセス実行システム400は、概説すれば、プロセスネットワーク生成部450がプロセスネットワークモデルDM4をシリアルライズして出力するプロセスネットワークモデル出力部452を有する点、およびプロセスネットワークモデル入力部451がシリアルライズされたプロセスネットワークモデルDM4の入力を受け付ける点において第2実施形態に係るプロセス実行システム200とは異なる。第4実施形態に係るプロセス実行システム400によれば、一のプロセス実行システム400において使用されているプロセスネットワークモデルDM4を他のプロセス実行システム400で使用することが容易になり、相互運用性が向上する。なお、第4実施形態に係るプロセス実行システム400は、以下で特に説明する構成を除いて、第2実施形態に係るプロセス実行システム200と同様に構成できる。
- [0364] 図62は、本実施形態に係るプロセス実行システム400のプロセスネットワーク生成部450の機能ブロック図である。図63は、本実施形態に係るプロセス実行システム400のプロセスネットワークモデル出力部452がプロセスネットワークモデルDM4をシリアルライズしたデータを例示する模式図である。以下、第4実施形態に係るプロセス実行システム400について説明する。
- [0365] 図62を参照して、本実施形態に係るプロセスネットワーク生成部450は、プロセスネットワークモデルDM4を出力するプロセスネットワークモデル出力部452をさらに有する。プロセスネットワークモデル出力部452は、プロセスネットワークモデルDM4をシリアルライズしたデータMDoutを出力する。
- [0366] 本実施形態に係るプロセスネットワークモデル入力部451は、プロセスネットワークモデルDM4をシリアルライズしたデータの入力を受け付ける。プロセスネットワーク生成部450は、プロセスネットワークモデル入力部451が入力を受け付けたプロセスネットワークモデルDM4をシリアルライズしたデータをデシリアルライズすることによって得られたプロセスネットワ

ークモデルDM4を用いてプロセスネットワークD4を生成する。

[0367] プロセスネットワークモデル出力部452は、プロセスネットワークモデルDM4をシリアルライズする際に、プロセスネットワークモデルDM4と、当該プロセスネットワークモデルDM4をシリアルライズしたデータMDoutが一対一に対応する規則に基づいてシリアルライズする。

[0368] プロセスネットワークモデルDM4と、当該プロセスネットワークモデルDM4をシリアルライズしたデータMDoutが一対一に対応するとは、プロセスネットワークモデルDM4が同一であればシリアルライズしたデータMDoutが同一となり、プロセスネットワークモデルDM4が異なればシリアルライズしたデータMDoutが異なることを意味する。逆に、シリアルライズしたデータMDoutが同一であればプロセスネットワークモデルDM4が同一となり、シリアルライズしたデータMDoutが異なればプロセスネットワークモデルDM4が異なるということでもある。

[0369] プロセスネットワークモデルDM4と、当該プロセスネットワークモデルDM4をシリアルライズしたデータMDoutが一対一に対応する規則は、例えば、プロセス番号マスタDM451の値を基準にプロセスマスタノードを昇順または降順に配列する規則とすることができる。また、階層型のプロセスネットワークモデルDM4の場合には、プロセスネットワークモデルDM4と、当該プロセスネットワークモデルDM4をシリアルライズして出力されるデータMDoutが一対一に対応する規則は、深さ優先、かつ、同一階層においてはプロセス番号マスタDM451の値を基準に昇順または降順で配列する規則とすることができる。

[0370] プロセスネットワークモデル出力部452がプロセスネットワークモデルDM4をシリアルライズしたデータMDoutを出力する際に使用するデータ形式は、当該シリアルライズしたデータMDoutをデシリアルライズすることによって、シリアルライズする対象となったプロセスネットワークモデルDM4を復元できる限りにおいて限定されない。

[0371] プロセスネットワークモデル出力部452がプロセスネットワークモデル

DM4 をシリアル化したデータMDout を出力する際に使用するデータ形式は、例えば、JSON、XML (Extensible Markup Language)、YAML または CSV であり得る。好ましくは、プロセスネットワークモデル出力部452 がプロセスネットワークモデルDM4 をシリアル化したデータMDout を出力する際に使用するデータ形式は、キーと値の組を配列したデータ形式である。さらに好ましくは、プロセスネットワークモデル出力部452 がプロセスネットワークモデルDM4 をシリアル化して出力する際に使用するデータ形式は、JSON (JavaScript Object Notation) である。

[0372] プロセスネットワークモデル出力部452 がプロセスネットワークモデルDM4 をシリアル化したデータMDout を出力する際に使用するデータ形式がキーと値の組を配列したデータ形式である場合、値は、文字列型、数値型、論理型、日付型、リスト型などの種々の型付きでデータを表現できるように構成できる。また、値は、キーと値の組を配列したデータを要素とするリストであるように構成できる。

[0373] 図63は、プロセスネットワークモデル出力部452 がプロセスネットワークモデルDM4 をシリアル化したデータMDout を例示する模式図である。図63は、プロセスネットワークモデル出力部452 がプロセスネットワークモデルDM4 をシリアル化したデータMDout を出力する際に使用するデータ形式がJSONである場合を例示している。図63は、プロセスマスタノードが実施形態1に係る図7に例示したデータ構造である場合を例示している。プロセスマスタノードが実施形態2に係る図24に例示したデータ構造である場合にも同様にシリアル化できる。

[0374] 図63を参照して、一のプロセスマスタノードDM40のデータは、「プロセス生成情報」と「子プロセスマスタノード群」をキーに有する。キーである「プロセス生成情報」に対応する値は、一のプロセスマスタノードDM40に含まれるプロセス生成情報をキーと値の組によって配列したデータである。キーである「子プロセスマスタノード群」に対応する値は、一のプロ

セスマスタノードDM40の直属の子のプロセスマスタノード群に含まれる
各々のプロセスマスタノードDM40のデータを配列したリストである。一
のセスマスタノードDM40に直属の子のプロセスマスタノードDM40
が存在しない場合には、キーである「子プロセスマスタノード群」に対応
する値に格納されるリストは空である。

[0375] (作用効果)

本実施形態に係るプロセス実行システム400のプロセスネットワーク生
成部450は、プロセスネットワークモデルDM4を出力するプロセスネッ
トワークモデル出力部452を有する。プロセスネットワークモデル出力部
452は、プロセスネットワークモデルDM4をシリアルライズして出力する
。

[0376] これにより、一のプロセス実行システム400において使用されているプ
ロセスネットワークモデルDM4を他のプロセス実行システム400で使用
することが容易になり、相互運用性が向上する。そのため、本実施形態に係
るプロセス実行システム400によれば、可塑性と安定性とを両立させるこ
とによって変化の速度が速いビジネス環境への適応度がより高いプロセス実
行システム400を提供できる。

[0377] なお、本実施形態に係るプロセス実行システム400によれば、プロセス
ネットワークモデルDM4においてプロセスとプロセスとの間に制御フロー
を表現するゲートを配置する必要がない。そのため、BPMNなどのゲート
を有する表記法で表現したモデルと比較して、プロセスネットワークモデル
DM4のシリアルライズが容易である。

[0378] 本実施形態に係るプロセス実行システム400のプロセスネットワークモ
デル入力部451は、プロセスネットワークモデルDM4をシリアルライズし
たデータの入力を受け付ける。プロセスネットワーク生成部450は、プロ
セスネットワークモデル入力部451が入力を受け付けたプロセスネットワ
ークモデルDM4をシリアルライズしたデータをデシリアルライズすることによ
って得られたプロセスネットワークモデルDM4を用いてプロセスネットワ

ークD4を生成する。

- [0379] これにより、一のプロセス実行システム400において使用されているプロセスネットワークモデルDM4を他のプロセス実行システム400に直接的に入力して使用できるようになり、相互運用性がより向上する。そのため、本実施形態に係るプロセス実行システム400によれば、可塑性と安定性とを両立させることによって変化の速度が速いビジネス環境への適応度がさらに高いプロセス実行システム400を提供できる。
- [0380] プロセス実行システム400を運用する上で、一のプロセスネットワークモデルDM4と他のプロセスネットワークモデルDM4との違いを容易に確認できるようにしておくことは非常に重要である。例えば、ビジネスプロセスが変更されたことに伴ってプロセスネットワークモデルDM4も変更されるため、プロセスネットワークモデルDM4はバージョン管理されるのが好ましい。その際に、バージョンアップ前とバージョンアップ後のプロセスネットワークモデルDM4の違いを容易に確認できるようにしておくことは、プロセス実行システム400を使用した業務の品質を確保する上で非常に重要である。
- [0381] 本実施形態に係るプロセス実行システム400のプロセスネットワークモデル出力部452は、プロセスネットワークモデルDM4をシリアルライズする際に、プロセスネットワークモデルDM4と、当該プロセスネットワークモデルDM4をシリアルライズしたデータが一对一に対応する規則に基づいてシリアルライズする。
- [0382] これにより、プロセスネットワークモデルDM4の違いを確認するのが非常に容易になる。例えば、バージョンアップ前後のプロセスネットワークモデルDM4をシリアルライズしたデータを汎用のテキストデータ比較ソフトなどを使用して比較すれば変更箇所を容易に確認できる。そのため、本実施形態に係るプロセス実行システム400によれば、可塑性と安定性とを両立させることによって変化の速度が速いビジネス環境への適応度がさらに高いプロセス実行システム400を提供できる。

- [0383] 一方で、BPMNなどのゲートを有する表記法の場合、特にゲートを追加・削除すると、モデルをシリアル化したデータが大きく変わってしまうことが多く、変更箇所を確認することが困難なことが多い。
- [0384] 本実施形態に係るプロセス実行システム400において、プロセスネットワークモデル出力部452がプロセスネットワークモデルDM4をシリアル化したデータMDoutを出力する際に使用するデータ形式は、キーと値の組を配列したデータ形式である。
- [0385] これにより、プロセスネットワークモデルDM4をシリアル化したデータの容量が小さくなるとともに可読性が向上する。そのため、本実施形態に係るプロセス実行システム400によれば、可塑性と安定性とを両立させることによって変化の速度が速いビジネス環境への適応度がさらに高いプロセス実行システム400を提供できる。
- [0386] 一方で、BPMNなどのゲートを有する表記法の場合、キーと値の組を配列したデータ形式では、アクティビティとゲートとの間の関係性を表現するのが難しい。そのため、BPMNなどのゲートを有する表記法の場合、モデルをシリアル化したデータが、XMLのように所定のタグで文字列を囲んで記述するデータ形式であることが多い。その結果、BPMNなどのゲートを有する表記法の場合、モデルをシリアル化したデータの容量が大きい上に、モデルをシリアル化したデータを人間が読んで理解するには困難なことが多い。
- [0387] 本実施形態に係るプロセス実行システム400において、プロセスネットワークモデル出力部452がプロセスネットワークモデルDM4をシリアル化したデータMDoutを出力する際に使用するデータ形式は、JSONである。
- [0388] JSONは、異なるシステム間、特に、インターネットなどのネットワークを介して接続された異なるシステム間においてデータを交換する際に現在広く使用されているデータ形式である。プロセスネットワークモデル出力部452がプロセスネットワークモデルDM4をシリアル化したデータを出

力する際に使用するデータ形式がJSONであることによって、一のプロセス実行システム400において使用されているプロセスネットワークモデルDM4を他のプロセス実行システム400で使うことがさらに容易になり、相互運用性がさらに向上する。そのため、本実施形態に係るプロセス実行システム400によれば、可塑性と安定性とを両立させることによって変化の速度が速いビジネス環境への適応度がさらに高いプロセス実行システム400を提供できる。

[0389] (第5の実施形態)

第5実施形態に係るプロセス実行システム500は、概説すれば、制御部510が、プロセスネットワークモデル表示部590と、プロセスネットワークD5を表示するプロセスネットワーク表示部595と、をさらに有する点において第2実施形態に係るプロセス実行システム200とは異なる。第5実施形態に係るプロセス実行システム500によれば、プロセスネットワークモデルDM5やプロセスネットワークD5を可視化できるため、プロセスネットワークモデルDM5やプロセスネットワークD5の複雑性が増しても業務の品質を確保するのがより容易になる。なお、第5実施形態に係るプロセス実行システム500は、以下で特に説明する構成を除いて、第2実施形態に係るプロセス実行システム200と同様に構成できる。

[0390] 図64は、本実施形態に係るプロセス実行システム500の制御部510の機能ブロック図である。図65は、本実施形態に係るプロセス実行システム500のプロセスネットワークモデル表示部590の機能ブロック図である。図66は、本実施形態に係るプロセス実行システム500のプロセスネットワーク表示部595の機能ブロック図である。図67は、本実施形態に係るプロセス実行システム500のプロセスネットワークモデル表示部590がプロセスネットワークモデルDM5を表示した様子を示す図である。図68は、本実施形態に係るプロセス実行システム500のプロセスネットワーク表示部595がプロセスネットワークD5を表示した様子を示す図である。以下、第5実施形態に係るプロセス実行システム500について説明す

る。

- [0391] 図64を参照して、本実施形態に係るプロセス実行システム500の制御部510は、プロセスネットワークモデルDM5を表示するプロセスネットワークモデル表示部590と、プロセスネットワークDM5を表示するプロセスネットワーク表示部595と、をさらに有する。
- [0392] 図65を参照して、プロセスネットワークモデル表示部590は、プロセスネットワークモデルDM5の表示要求を受け付けるプロセスネットワークモデル表示要求受付部591と、当該表示要求の要求元においてプロセスネットワークモデルDM5を表示するためのプロセスネットワークモデル表示データを生成するプロセスネットワークモデル表示データ生成部592と、を有する。
- [0393] プロセスネットワークモデル表示部590は、プロセスネットワークモデル表示要求受付部591が表示要求を受け付けると、プロセスネットワークモデル表示データ生成部592においてプロセスネットワークモデル表示データを生成して、当該表示要求の要求元にプロセスネットワークモデル表示データを返信する。
- [0394] プロセスネットワークモデル表示要求受付部591が表示要求を受け付ける方法は特に限定されない。例えば、HTTPを使用した通信によってプロセスネットワークモデルDM5の表示要求を受け付けることができる。
- [0395] プロセスネットワークモデル表示データ生成部592は、プロセスネットワークモデルDM5の表示要求の要求元においてプロセスネットワークモデルDM5を3次元表示するためのプロセスネットワークモデル3次元表示データを生成する。
- [0396] プロセスネットワークモデル3次元表示データは、プロセスネットワークモデルDM5に含まれるプロセスマスタノードDM50をノードとし、プロセスマスタノードDM50同士の関係をエッジとする3次元のネットワーク構造によってプロセスネットワークモデルDM5を表現する。階層型のプロセスネットワークモデルDM5の場合、プロセスネットワークモデル3次元

表示データは、プロセスネットワークモデルDM5に含まれるプロセスマスタノードDM50をノードとし、プロセスマスタノードDM50同士の親子関係をエッジとする3次元のネットワーク構造によってプロセスネットワークモデルDM5を表現する。

[0397] プロセスネットワークモデル3次元表示データを生成する方法は特に限定されない。例えば、プログラミング言語Pythonから使用可能なNetworkXなどのライブラリを用いて、プロセスマスタノードDM50に3次元座標を割り当てることによって、プロセスネットワークモデル3次元表示データを生成できる。

[0398] プロセスネットワークモデルDM5の表示要求には、プロセスネットワークモデルDM5の表示要求の要求元におけるプロセスネットワークモデルDM5の表示形態を制御するプロセスネットワークモデル表示制御情報が含まれる。

[0399] プロセスネットワークモデル表示データ生成部592は、プロセスネットワークモデルDM5の表示要求の要求元におけるプロセスネットワークモデルDM5の表示形態がプロセスネットワークモデル表示制御情報に応じて変化するように、プロセスネットワークモデル表示データを生成する。

[0400] プロセスネットワークモデル表示制御情報に応じたプロセスネットワークモデルDM5の表示形態の変化は、例えば、プロセスネットワークモデルDM5の回転、拡大、縮小および平行移動からなる群から選択される少なくとも1つ以上の表示形態の変化の組み合わせである。プロセスネットワークモデル表示制御情報に応じたプロセスネットワークモデルDM5の表示形態の変化は、プロセスネットワークモデルDM5に含まれるプロセスマスタノードDM50に関する情報の表示および非表示であってもよい。

[0401] プロセスネットワークモデル表示要求受付部591がプロセスネットワークモデル表示制御情報を含むプロセスネットワークモデルDM5の表示要求を受け付ける方法は特に限定されない。例えば、プログラミング言語Pythonから使用可能なPlotlyなどのライブラリを用いることによって

、プロセスネットワークモデル表示要求受付部591がプロセスネットワークモデル表示制御情報を含むプロセスネットワークモデルDM5の表示要求を受け付けることができる。

[0402] プロセスネットワークモデル表示部590がプロセスネットワークモデルDM5の表示要求の要求元にプロセスネットワークモデル表示データを返信する方法は特に限定されない。例えば、HTTPを使用した通信によってプロセスネットワークモデル表示データを返信するように構成できる。

[0403] プロセスネットワークモデル表示部590は、プロセスネットワークモデルDM5に含まれるプロセスマスタノードDM50に関する情報をプロセスマスタノードDM50ごとに表示してもよい。プロセスネットワークモデル表示部590は、プロセスネットワークモデルDM5に含まれるプロセスマスタノードDM50に関する情報をノードの色や形で表現してもよい。例えば、プロセスネットワークモデル表示部590は、プロセスマスタノードDM50のプロセス生成情報に含まれる非同期フラグマスタDM554や自動起動フラグマスタDM555の値に応じて、ノードの色や形を変更してもよい。

[0404] 図66を参照して、プロセスネットワーク表示部595は、プロセスネットワークD5の表示要求を受け付けるプロセスネットワーク表示要求受付部596と、当該表示要求の要求元においてプロセスネットワークD5を表示するためのプロセスネットワーク表示データを生成するプロセスネットワーク表示データ生成部597と、を有する。

[0405] プロセスネットワーク表示部595は、プロセスネットワーク表示要求受付部596がプロセスネットワークD5の表示要求を受け付けると、プロセスネットワーク表示データ生成部597においてプロセスネットワーク表示データを生成して、当該表示要求の要求元にプロセスネットワーク表示データを返信する。

[0406] プロセスネットワーク表示要求受付部596がプロセスネットワークD5の表示要求を受け付ける方法は特に限定されず、プロセスネットワークモデ

ル表示要求受付部591と同様に構成できる。

- [0407] プロセスネットワーク表示データ生成部597は、プロセスネットワークD5の表示要求の要求元においてプロセスネットワークD5を3次元表示するためのプロセスネットワーク3次元表示データを生成する。
- [0408] プロセスネットワーク3次元表示データは、プロセスネットワークD5に含まれるプロセスノードD50をノードとし、プロセスノードD50同士の関係をエッジとする3次元のネットワーク構造によってプロセスネットワークD5を表現するデータである。階層型のプロセスネットワークD5の場合、プロセスネットワーク3次元表示データは、プロセスネットワークD5に含まれるプロセスノードD50をノードとし、プロセスノードD50同士の親子関係をエッジとする3次元のネットワーク構造によってプロセスネットワークD5を表現するデータである。
- [0409] プロセスネットワーク3次元表示データを生成する方法は特に限定されず、プロセスネットワークモデル3次元表示データを生成する方法と同様に構成できる。
- [0410] プロセスネットワークD5の表示要求には、プロセスネットワークD5の表示要求の要求元におけるプロセスネットワークD5の表示形態を制御するプロセスネットワーク表示制御情報が含まれる。
- [0411] プロセスネットワーク表示データ生成部597は、プロセスネットワークD5の表示要求の要求元におけるプロセスネットワークD5の表示形態がプロセスネットワーク表示制御情報に応じて変化するように、プロセスネットワーク表示データを生成する。
- [0412] プロセスネットワーク表示制御情報に応じたプロセスネットワークD5の表示形態の変化は、例えば、プロセスネットワークD5の回転、拡大、縮小および平行移動からなる群から選択される少なくとも1つ以上の表示形態の変化の組み合わせである。プロセスネットワーク表示制御情報に応じたプロセスネットワークD5の表示形態の変化は、プロセスネットワークD5に含まれるプロセスノードD50に関する情報の表示および非表示であってもよ

い。

- [0413] プロセスネットワーク表示要求受付部596がプロセスネットワーク表示制御情報を含むプロセスネットワークD5の表示要求を受け付ける方法は特に限定されず、プロセスネットワークモデル表示要求受付部591と同様に構成できる。
- [0414] プロセスネットワーク表示部595がプロセスネットワークD5の表示要求の要求元にプロセスネットワーク表示データを返信する方法は特に限定されず、プロセスネットワークモデル表示部590と同様に構成できる。
- [0415] プロセスネットワーク表示部595は、プロセスネットワークD5に含まれるプロセスノードD50に関する情報をプロセスノードD50ごとに表示してもよい。プロセスネットワーク表示部595は、プロセスネットワークD5に含まれるプロセスノードD50に関する情報をノードの色や形で表現してもよい。例えば、プロセスネットワーク表示部595は、プロセスノードD50のプロセス状態D53に応じて、ノードの色や形を変更してもよい。
- [0416] 図67は、ブラウザからHTTP通信によってプロセスネットワークモデル表示部590にプロセスネットワークモデルDM5の表示要求を行って、プロセスネットワークモデル表示部590から返信されたプロセスネットワークモデル3次元表示データによって表現されるプロセスネットワークモデルDM5をブラウザに3次元表示した様子を示す図である。ブラウザに表示したプロセスネットワークモデルDM5は、ブラウザ上のマウス操作によって、回転、拡大、縮小および平行移動させることが可能である。また、ブラウザに表示したプロセスネットワークモデルDM5に含まれるプロセスマスタノードDM50にマウスを合わせると、当該プロセスマスタノードDM50のプロセス生成情報をポップアップによって表示することもできる。
- [0417] 図68は、ブラウザからHTTP通信によってプロセスネットワーク表示部595にプロセスネットワークD5の表示要求を行って、プロセスネットワーク表示部595から返信されたプロセスネットワーク3次元表示データ

によって表現されるプロセスネットワークD5をブラウザに3次元表示した様子を示す図である。ブラウザに表示したプロセスネットワークD5は、ブラウザ上のマウス操作によって、回転、拡大、縮小および平行移動させることが可能である。また、ブラウザに表示したプロセスネットワークD5に含まれるプロセスノードD50にマウスを合わせると、当該プロセスノードD50のプロセス定義情報をポップアップによって表示することもできる。

[0418] (作用効果)

プロセスネットワークモデルDM5の可視化は、プロセス実行システム500を使用した業務の品質を確保する上で非常に重要である。特に、プロセスネットワークモデルDM5に含まれるプロセスマスタノードDM50の数が増えてモデルの複雑性が増した際に、プロセスネットワークモデルDM5の可視化は、プロセス実行システム500を使用した業務の品質を確保する上で非常に重要な役割を果たす。

[0419] 本実施形態に係るプロセス実行システム500は、プロセスネットワークモデルDM5を表示するプロセスネットワークモデル表示部590をさらに有する。

[0420] これにより、プロセスネットワークモデルDM5に含まれるプロセスマスタノードDM50の数が増えてモデルの複雑性が増しても、プロセス実行システム500を使用した業務の品質を確保するのがより容易になる。そのため、本実施形態に係るプロセス実行システム500によれば、可塑性と安定性とを両立させることによって変化の速度が速いビジネス環境への適応度がさらに高いプロセス実行システム500を提供できる。

[0421] 一方で、BPMNなどのゲートを有する表記法の場合、モデルを可視化した際にゲートとアクティビティが混在して表示されるため、モデルの検証を行うのが容易ではないことが多い。特に、階層化されたモデルや多くのアクティビティやゲートを含む複雑なモデルを可視化しても視認性が悪く、モデルの検証が難しいことが多い。

[0422] また、本実施形態に係るプロセス実行システム500のプロセスネットワ

ークモデル表示部590は、プロセスネットワークモデルDM5の表示要求を受け付けるプロセスネットワークモデル表示要求受付部591と、当該表示要求の要求元においてプロセスネットワークモデルDM5を表示するためのプロセスネットワークモデル表示データを生成するプロセスネットワークモデル表示データ生成部592と、を有する。プロセスネットワークモデル表示データ生成部592は、プロセスネットワークモデルDM5の表示要求の要求元においてプロセスネットワークモデルDM5を3次元表示するためのプロセスネットワークモデル3次元表示データを生成する。

[0423] これにより、プロセスネットワークモデルDM5に含まれるプロセスマスタノードDM50の数が増えてモデルの複雑性が増しても、プロセス実行システム500を使用した業務の品質を確保するのがさらに容易になる。そのため、本実施形態に係るプロセス実行システム500によれば、可塑性と安定性とを両立させることによって変化の速度が速いビジネス環境への適応度がさらに高いプロセス実行システム500を提供できる。

[0424] また、本実施形態に係るプロセス実行システム500のプロセスネットワークモデル表示部590は、プロセスネットワークモデルDM5に含まれるプロセスマスタノードDM50に関する情報をプロセスマスタノードDM50ごとに表示する。

[0425] これにより、プロセスネットワークモデルDM5に含まれるプロセスマスタノードDM50の数が増えてモデルの複雑性が増しても、プロセスモデルを容易に検証できる。そのため、本実施形態に係るプロセス実行システム500によれば、可塑性と安定性とを両立させることによって変化の速度が速いビジネス環境への適応度がさらに高いプロセス実行システム300を提供できる。

[0426] プロセスネットワークD5の可視化は、プロセス実行システム500を使用した業務の品質を確保する上で非常に重要である。特に、プロセスネットワークD5に含まれるプロセスノードD50の数が増えてモデルの複雑性が増した際に、プロセスネットワークD5の可視化は、プロセス実行システム

５００を使用した業務の品質を確保する上で非常に重要な役割を果たす。

[0427] 本実施形態に係るプロセス実行システム５００は、プロセスネットワークＤ５を表示するプロセスネットワーク表示部５９５をさらに有する。

[0428] これにより、プロセスネットワークＤ５に含まれるプロセスノードＤ５０の数が増えてモデルの複雑性が増しても、プロセス実行システム５００を使用した業務の品質を確保するのがさらに容易になる。そのため、本実施形態に係るプロセス実行システム５００によれば、可塑性と安定性とを両立させることによって変化の速度が速いビジネス環境への適応度がさらに高いプロセス実行システム５００を提供できる。

[0429] また、本実施形態に係るプロセス実行システム５００のプロセスネットワーク表示部５９５は、プロセスネットワークＤ５の表示要求を受け付けるプロセスネットワーク表示要求受付部５９６と、当該表示要求の要求元においてプロセスネットワークＤ５を表示するためのプロセスネットワーク表示データを生成するプロセスネットワーク表示データ生成部５９７と、を有する。プロセスネットワーク表示データ生成部５９７は、プロセスネットワークＤ５の表示要求の要求元においてプロセスネットワークＤ５を３次元表示するためのプロセスネットワーク３次元表示データを生成する。

[0430] これにより、プロセスネットワークＤ５に含まれるプロセスノードＤ５０の数が増えてモデルの複雑性が増しても、プロセス実行システム５００を使用した業務の品質を確保するのがさらに容易になる。そのため、本実施形態に係るプロセス実行システム５００によれば、可塑性と安定性とを両立させることによって変化の速度が速いビジネス環境への適応度がさらに高いプロセス実行システム５００を提供できる。

[0431] また、本実施形態に係るプロセス実行システム５００のプロセスネットワーク表示部５９５は、プロセスネットワークＤ５に含まれるプロセスノードＤ５０に関する情報をプロセスノードＤ５０ごとに表示する。

[0432] これにより、プロセスネットワークＤ５に含まれるプロセスノードＤ５０の数が増えてモデルの複雑性が増しても、プロセスの実行状況のモニタリン

グが容易になる。そのため、本実施形態に係るプロセス実行システム500によれば、可塑性と安定性とを両立させることによって変化の速度が速いビジネス環境への適応度がさらに高いプロセス実行システム300を提供できる。

[0433] (実施例)

本発明の作用効果を、実施例を用いてさらに具体的に説明する。ただし、本発明の技術的範囲が本実施例のみに制限されるわけではない。本実施例は、上述した第2実施形態に係るプロセス実行システム200の構成をもとに説明するが、上述した第1実施形態に係るプロセス実行システム100、実施形態3に係るプロセス実行システム300、実施形態4に係るプロセス実行システム400、および実施形態5に係るプロセス実行システム500の作用効果についても本実施例をもとに理解できる。

[0434] 本実施例に係るプロセス実行システム200は、オンプレミスサーバー上で動作する。本実施例に係るプロセス実行システム200の制御部210は、オンプレミスサーバーが有するCPU (Central Processing Unit)、およびRAM (Random Access Memory)、ROM (Read Only Memory)等のメモリにより構成される。本実施例に係るプロセス実行システム200の記憶部220は、オンプレミスサーバーが有するSSD (Solid State Drive)等により構成される。本実施例に係るプロセス実行システム200の通信部230は、オンプレミスサーバーが有するインターフェース回路 (LANカード等)で構成される。

[0435] 本実施例において使用するアクション実行システム50は、Microsoft CorporationのクラウドサービスであるMICROSOFT 365 (登録商標)のPower AutomateおよびPower Automate for desktopによって構成される。

[0436] 本実施例に係るプロセス実行システム200の制御部210は、プロセスネットワーク生成部250、プロセスネットワーク状態管理部260、プロ

セスネットワーク実行部270として機能する。プロセスネットワーク生成部250、プロセスネットワーク状態管理部260およびプロセスネットワーク実行部270は、Webアプリ作成用のフレームワークであるDjangoを使用して構築されたWebアプリの機能として実装される。

[0437] 本実施例に係るプロセス実行システム200の記憶部220は、データベースサーバーとして機能する。データベースサーバーは、PostgreSQLを使用して構成される。プロセスネットワークモデルDM1およびプロセスネットワークD1はデータベースサーバーとして機能する記憶部220に保存される。

[0438] 本実施例に係るプロセス実行システム200の通信部230は、クラウドサービスとの通信を実行するゲートウェーサーバーとして機能する。これにより、オンプレミスサーバーとクラウドサービスとの間で通信を行うことができる。本実施例では、Microsoft Corporationが提供するon-premises data gatewayを使用して、通信部130をゲートウェーサーバーとして機能させる。on-premises data gatewayを使用することによって、アクション実行システム50として機能するPower Automateからプロセス実行システム200に対してAPI経由でアクセスできる。

[0439] プロセスネットワーク生成部250のプロセスネットワークモデル入力部151は、WebブラウザからアクセスすることによってプロセスネットワークモデルDM2の登録や操作を行うことが可能なGUI(Graphical User Interface)を有する。

[0440] プロセスネットワーク状態管理部260は、Webアプリを構成するライブラリの一部として構成されている。プロセスネットワーク状態管理部260を構成するライブラリは、Microsoft CorporationのオペレーティングシステムWindows(登録商標)のタスクスケジューラによって定期的に起動するタスクから実行される。これにより、プロセスネットワーク状態管理部260は、プロセスネットワークD2に含まれる

プロセスノードD20の状態遷移処理を定期的に行うことができる。また、プロセスネットワーク状態管理部260は、プロセスの状態を変更する状態変更要求を受け付けるAPIを有する。当該APIを使用することによって、アクション実行システム50から特定のプロセスノードD20の状態を変更することができる。

[0441] プロセスネットワーク実行部270は、Webアプリを構成するライブラリの一部として構成されている。プロセスネットワーク実行部270を構成するライブラリは、Microsoft CorporationのオペレーティングシステムWindows（登録商標）のタスクスケジューラによって定期的に起動するタスクから実行できる。これにより、プロセスネットワーク実行部270は、実行可能なプロセスノードD20の実行処理を定期的に実行できる。

[0442] 本実施例では、プロセスネットワーク実行部270は、実行可能なプロセスノードD20の実行処理を開始する際に、プロセスノードD20のプロセス状態D23を実行中状態にする。また、プロセスネットワーク実行部270は、非同期フラグD254がTrueのプロセスノードD20の実行処理を完了した際に、プロセスノードD20のプロセス状態D23を待機中状態にする。また、プロセスネットワーク状態管理部260は、プロセスノードD20を実行可能状態にするステップS1326（図13参照）およびステップS13245（図14参照）において、プロセスノードD20のプロセス状態D23が実行中状態または待機中状態であった場合には、当該プロセスノードD20のプロセス状態D23は変化させない。

[0443] 本実施例では、プロセス実行システム200を使用して、特許庁に対して特許年金を納付する業務に係るビジネスプロセスを処理する。

[0444] 特許年金を納付する対象の案件のコンテキストデータは、データベースサーバーとして機能する記憶部220に保存されている。特許年金を納付する対象の案件のコンテキストデータは、出願番号や出願人、請求項の数、出願人毎の持ち分の割合、および特許料金の軽減有無などの情報を含む。特許年

金を納付する対象の案件のコンテキストデータは、案件毎に一意に設定された案件番号を主キーにしてデータベースサーバーとして機能する記憶部220から取得することができる。

[0445] データベースサーバーとして機能する記憶部220には、アクション実行システム50として機能するPower AutomateやPower Automate for desktopから、ゲートウェーサーバーとして機能する通信部230を介してアクセス可能である。Power AutomateやPower Automate for desktopのSQLクエリを送信するアクション等を使用することで、案件番号をキーにして特定の案件のコンテキストデータをデータベースサーバーとして機能する記憶部220から通信部230を介して取得できる。

[0446] 案件番号等のコンテキストデータは、プロセスネットワーク生成部250がプロセスネットワークモデルDM1からコンテキストデータを使用してプロセスネットワークD1を生成する際に、アクションアイテムのAPIデータに埋め込む。APIデータに対する埋め込みは、Djangoのテンプレートエンジンで使用するDjango Template Languageと同様の仕組みで行う。すなわち、APIデータに対する埋め込みは、プロセスネットワークモデルDM1のアクションアイテムマスタDM1531のAPIデータマスタに埋め込まれたタグをコンテキストデータに基づいて置換することで行う。

[0447] 図69から図73を参照して、本実施例で使用するプロセスネットワークモデルDM2およびプロセスネットワークD2について説明する。図69から図73は、本実施例に係るプロセス実行システムにおいて実行される処理について説明するために使用するプロセスネットワークモデルおよびプロセスネットワークの一部のデータ構造を示す模式図である。プロセスネットワーク生成部250は、プロセスネットワークモデルDM2に含まれるプロセスマスタノードと一対一に対応するプロセスノードを含むプロセスネットワークD2を生成する。

- [0448] 図69を参照して、本実施例で使用するプロセスネットワークモデルDM2は、ルートプロセスマスタノードML01を有する。ルートプロセスマスタノードML01は、ルートプロセスノードL01を生成するのに使用される。
- [0449] ルートプロセスマスタノードML01は、プロセスマスタノードML11、ML12、ML13およびML14を有する。プロセスマスタノードML11は、問い合わせプロセスノードL11を生成するのに使用される。プロセスマスタノードML12は、指示受領プロセスノードL12を生成するのに使用される。プロセスマスタノードML13は、庁手続プロセスノードL13を生成するのに使用される。プロセスマスタノードML14は、請求プロセスノードL14を生成するのに使用される。
- [0450] 図70を参照して、プロセスマスタノードML11は、プロセスマスタノードML21、ML22およびML23を有する。プロセスマスタノードML21は、問い合わせメール生成プロセスノードL21を生成するのに使用される。プロセスマスタノードML22は、問い合わせメール送信プロセスノードL22を生成するのに使用される。プロセスマスタノードML13は、ドキュメント生成プロセス起動プロセスノードL13を生成するのに使用される。
- [0451] 図71を参照して、プロセスマスタノードML12は、プロセスマスタノードML24およびML25を有する。プロセスマスタノードML24は、指示確認プロセスノードL24を生成するのに使用される。プロセスマスタノードML25は、ドキュメント送信プロセス起動プロセスノードL25を生成するのに使用される。
- [0452] 図72を参照して、プロセスマスタノードML13は、プロセスマスタノードML26およびML27を有する。プロセスマスタノードML26は、庁書類生成プロセスノードL26を生成するのに使用される。プロセスマスタノードML27は、庁書類提出プロセスノードL27を生成するのに使用される。

- [0453] 図73を参照して、プロセスマスタノードML14は、プロセスマスタノードML28およびML29を有する。プロセスマスタノードML28は、請求書生成プロセスノードL28を生成するのに使用される。プロセスマスタノードML29は、請求書送付プロセスノードL29を生成するのに使用される。
- [0454] 以下、各プロセスノードについて詳説する。
- [0455] 図70を参照して、問い合わせプロセスノードL11は、問い合わせメール生成プロセスノードL21、問い合わせメール送信プロセスノードL22およびドキュメント生成プロセス起動プロセスノードL23を子として有する。
- [0456] プロセスノードL21は、問い合わせメール生成プロセスを表現する。問い合わせメール生成プロセスでは、特許年金を納付する意思があるか否かを出願人に問い合わせるためのメールを生成する。問い合わせメールの生成は、Power Automateに登録した問い合わせメール生成用のフローを実行することによって行う。当該問い合わせメール生成用のフローの実行は、Power AutomateのAPIに当該フローの実行要求を送信することによって行う。問い合わせメール生成用のフローは、問い合わせメールの件名や本文の生成、宛先の設定、必要に応じて添付ファイルの添付等を行う。
- [0457] プロセスノードL21のアクション情報D253は、Power Automateに登録した問い合わせメール生成用のフローを実行するためのAPIアドレスを含む。また、プロセスノードL21のアクション情報D253は、案件番号等のコンテキストデータをAPIデータとして含む。プロセスネットワーク実行部270は、プロセスノードL21のアクション情報D253に含まれるAPIアドレスに対して、問い合わせメール生成用のフローの実行要求を送信するとともに、当該アクション情報D253に含まれるAPIデータを送信する。問い合わせメール生成用のフローは、送信されたAPIデータに含まれるコンテキストデータを使用して、データベースサー

バーとして機能する記憶部220から必要な情報を取得し、メールの宛先、件名、本文の内容および添付ファイルの内容等を動的に変化させる。

[0458] プロセスノードL22は、問い合わせメール送信プロセスを表現する。問い合わせメール送信プロセスでは、問い合わせメールの送信依頼を業務担当者に行う。問い合わせメールの送信依頼は、Power Automateに登録した問い合わせメール送信依頼用のフローを実行することによって行う。当該問い合わせメール送信依頼用のフローの実行は、Power AutomateのAPIに当該フローの実行要求を送信することによって行う。問い合わせメール送信依頼用のフローは、MICROSOFT 365（登録商標）のタスク管理サービスであるPlannerに、問い合わせメール送信タスクを登録し、当該登録したタスクに業務担当者を割り当てる。業務担当者にタスクが割り当てられると、業務担当者に通知される。

[0459] プロセスノードL22のアクション情報D253は、Power Automateに登録した問い合わせメール送信依頼用のフローを実行するためのAPIアドレスを含む。また、プロセスノードL22のアクション情報D253は、案件番号等のコンテキストデータをAPIデータとして含む。プロセスネットワーク実行部270は、プロセスノードL22のアクション情報D253に含まれるAPIアドレスに対して、問い合わせメール送信依頼用のフローの実行要求を送信するとともに、当該アクション情報D253に含まれるAPIデータを送信する。問い合わせメール送信依頼用のフローは、送信されたAPIデータに含まれるコンテキストデータを使用して、データベースサーバーとして機能する記憶部220から必要な情報を取得し、問い合わせメール送信タスクに設定するメッセージの内容や、当該タスクを割り振る業務担当者を動的に変化させる。

[0460] 問い合わせメール送信タスクが業務担当者に割り当てられると、当該業務担当者に通知される。通知を受信した業務担当者は、問い合わせメール送信タスクに設定された内容に基づいて、上述した問い合わせメール生成用のフローが生成した問い合わせメールを送信する。業務担当者は、問い合わせメ

ールの送信を完了すると、P l a n n e r に登録された問い合わせメール送信タスクを完了する。問い合わせメール送信タスクが完了すると、当該タスクの完了をトリガとしてP o w e r A u t o m a t e に登録した問い合わせメール送信プロセスノード完了処理用のフローが起動する。問い合わせメール送信プロセスノード完了処理用のフローは、プロセスネットワーク状態管理部260のAPIにプロセスノードL22のプロセス状態D23を完了状態に変更するプロセス状態変更要求を送信する。

[0461] プロセスノードL23は、ドキュメント生成プロセスを起動するプロセスを表現する。ドキュメント生成プロセスを起動するプロセスでは、庁書類生成プロセスを表現するプロセスノードL26のプロセス状態D23を実行可能状態にするとともに、請求書生成プロセスを表現するプロセスノードL28のプロセス状態D23を実行可能状態にする。ドキュメント生成プロセスの起動は、P o w e r A u t o m a t e に登録したドキュメント生成プロセス起動用のフローを実行することによって行う。当該ドキュメント生成プロセス起動用のフローの実行は、P o w e r A u t o m a t e のAPIに当該フローの実行要求を送信することによって行う。ドキュメント生成プロセス起動用のフローは、プロセスネットワーク状態管理部260のAPIにプロセスノードL26のプロセス状態D23を実行可能状態に変更するプロセス状態変更要求を送信するとともに、プロセスノードL28のプロセス状態D23を実行可能状態に変更するプロセス状態変更要求を送信する。

[0462] プロセスノードL23のアクション情報D253は、P o w e r A u t o m a t e に登録したドキュメント生成プロセス起動用のフローを実行するためのAPIアドレスを含む。また、プロセスノードL23のアクション情報D253は、案件番号等のコンテキストデータをAPIデータとして含む。プロセスネットワーク実行部270は、プロセスノードL23のアクション情報D253に含まれるAPIアドレスに対して、ドキュメント生成プロセス起動用のフローの実行要求を送信するとともに、当該アクション情報D253に含まれるAPIデータを送信する。ドキュメント生成プロセス起動

用のフローは、送信されたAPIデータに含まれるコンテキストデータを使用して、データベースサーバーとして機能する記憶部220から、実行可能状態にするプロセスノードL26およびL28を特定するのに必要な情報（プロセスノード識別子D21等）を取得する。

[0463] 図71を参照して、指示受領プロセスノードL12は、指示確認プロセスノードL24およびドキュメント送信プロセス起動プロセスノードL25を子として有する。

[0464] プロセスノードL24は、指示確認プロセスを表現する。指示確認プロセスでは、指示メールの確認依頼を業務担当者に行う。指示メールの確認依頼は、Power Automateに登録した指示メール確認依頼用のフローを実行することによって行う。当該指示メール確認依頼用のフローの実行は、Power AutomateのAPIに当該フローの実行要求を送信することによって行う。指示メール確認依頼用のフローは、MICROSOFT 365（登録商標）の承認依頼サービスを使用して、指示メール確認依頼のための承認依頼を業務担当者に対して送信する。

[0465] プロセスノードL24のアクション情報D253は、Power Automateに登録した指示メール確認依頼用のフローを実行するためのAPIアドレスを含む。また、プロセスノードL24のアクション情報D253は、案件番号等のコンテキストデータをAPIデータとして含む。プロセスネットワーク実行部270は、プロセスノードL24のアクション情報D253に含まれるAPIアドレスに対して、指示メール確認依頼用のフローの実行要求を送信するとともに、当該アクション情報D253に含まれるAPIデータを送信する。指示メール送信依頼用のフローは、送信されたAPIデータに含まれるコンテキストデータを使用して、データベースサーバーとして機能する記憶部220から必要な情報を取得し、指示メール確認依頼のための承認依頼に設定するメッセージの内容や、当該承認依頼を送信する業務担当者を動的に変化させる。

[0466] 指示確認プロセスを表現するプロセスノードL24は、プロセスノードL

22において送信した問い合わせメールに対する出願人からの指示メールを受信することによって実行可能状態になる。具体的には、問い合わせメールに対する出願人からの指示メールを受信すると、当該メールの受信をトリガとしてPower Automateに登録した指示確認プロセス起動処理用のフローが起動する。指示確認プロセス起動処理用のフローは、プロセスネットワーク状態管理部260のAPIにプロセスノードL24のプロセス状態D23を実行可能状態に変更するプロセス状態変更要求を送信する。

[0467] 上述したように、指示確認プロセスを表現するプロセスノードL24が実行されると、指示メール確認依頼のための承認依頼が業務担当者に送信される。指示メール確認依頼のための承認依頼を受け取った業務担当者は、当該承認依頼に設定された内容に基づいて、上述した問い合わせメール生成用のフローが生成した問い合わせメールに対する出願人からの指示メールを確認する。業務担当者は、指示メールを確認したら、指示メール確認依頼のための承認依頼を完了する。指示メール確認依頼のための承認依頼が完了すると、承認依頼の完了をトリガとしてPower Automateに登録した指示確認プロセスノード完了処理用のフローが起動する。指示確認プロセスノード完了処理用のフローは、プロセスネットワーク状態管理部160のAPIにプロセスノードL24のプロセス状態D23を完了状態に変更するプロセス状態変更要求を送信する。

[0468] プロセスノードL25は、ドキュメント送信プロセスを起動するプロセスを表現する。ドキュメント送信プロセスを起動するプロセスでは、庁書類提出プロセスを表現するプロセスノードL27のプロセス状態D23を実行可能状態にするとともに、請求書送付プロセスを表現するプロセスノードL29のプロセス状態D23を実行可能状態にする。ドキュメント提出プロセスの起動は、Power Automateに登録したドキュメント提出プロセス起動用のフローを実行することによって行う。当該ドキュメント提出プロセス起動用のフローの実行は、Power AutomateのAPIに当該フローの実行要求を送信することによって行う。ドキュメント提出プロ

セス起動用のフローは、プロセスネットワーク状態管理部260のAPIにプロセスノードL27のプロセス状態D23を実行可能状態に変更するプロセス状態変更要求を送信するとともに、プロセスノードL29のプロセス状態D23を実行可能状態に変更するプロセス状態変更要求を送信する。

[0469] プロセスノードL25のアクション情報D253は、Power Automateに登録したドキュメント提出プロセス起動用のフローを実行するためのAPIアドレスを含む。また、プロセスノードL25のアクション情報D253は、案件番号等のコンテキストデータをAPIデータとして含む。プロセスネットワーク実行部270は、プロセスノードL25のアクション情報D253に含まれるAPIアドレスに対して、問い合わせメール送信依頼用のフローの実行要求を送信するとともに、当該アクション情報D253に含まれるAPIデータを送信する。ドキュメント提出プロセス起動用のフローは、送信されたAPIデータに含まれるコンテキストデータを使用して、データベースサーバーとして機能する記憶部220から、実行可能状態にするプロセスノードL27およびL29を特定するのに必要な情報（プロセスノード識別子D21等）を取得する。

[0470] 図72を参照して、庁手続プロセスノードL13は、庁書類生成プロセスノードL26および庁書類提出プロセスノードL27を子として有する。

[0471] プロセスノードL26は、庁書類生成プロセスを表現する。庁書類生成プロセスでは、特許庁に提出する年金納付書を生成する。年金納付書の生成は、Power Automateに登録した年金納付書生成用のフローを実行することによって行う。当該年金納付書生成用のフローの実行は、Power AutomateのAPIに当該フローの実行要求を送信することによって行う。年金納付書生成用のフローは、年金納付書のフォーマットに基づいて各欄の情報をHTMLファイルに出力することによって年金納付書を生成する。

[0472] プロセスノードL26のアクション情報D253は、Power Automateに登録した年金納付書生成用のフローを実行するためのAPIア

ドレスを含む。また、プロセスノードL 2 6のアクション情報D 2 5 3は、案件番号等のコンテキストデータをAPIデータとして含む。プロセスネットワーク実行部2 7 0は、プロセスノードL 2 6のアクション情報D 2 5 3に含まれるAPIアドレスに対して、年金納付書生成用のフローの実行要求を送信するとともに、当該アクション情報D 2 5 3に含まれるAPIデータも送信する。年金納付書生成用のフローは、送信されたAPIデータに含まれるコンテキストデータを使用して、データベースサーバーとして機能する記憶部2 2 0から必要な情報を取得し、年金納付書の各欄（出願番号、特許出願人、請求項の数、納付年度、特許料など）の情報等を動的に変化させる。

[0473] プロセスノードL 2 7は、庁書類提出プロセスを表現する。庁書類提出プロセスでは、庁書類提出依頼を業務担当者に行う。庁書類提出依頼は、Power Automateに登録した庁書類提出依頼用のフローを実行することによって行う。当該庁書類提出依頼用のフローの実行は、Power AutomateのAPIに当該フローの実行要求を送信することによって行う。庁書類提出依頼用のフローは、MICROSOFT 365（登録商標）のタスク管理サービスであるPlannerに、庁書類提出タスクを登録し、当該登録したタスクに業務担当者を割り当てる。業務担当者にタスクが割り当てられると、業務担当者に通知される。

[0474] プロセスノードL 2 7のアクション情報D 2 5 3は、Power Automateに登録した庁書類提出依頼用のフローを実行するためのAPIアドレスを含む。また、プロセスノードL 2 7のアクション情報D 2 5 3は、案件番号等のコンテキストデータをAPIデータとして含む。プロセスネットワーク実行部2 7 0は、プロセスノードL 2 7のアクション情報D 2 5 3に含まれるAPIアドレスに対して、庁書類提出依頼用のフローの実行要求を送信するとともに、当該アクション情報D 2 5 3に含まれるAPIデータも送信する。庁書類提出依頼用のフローは、送信されたAPIデータに含まれるコンテキストデータを使用して、データベースサーバーとして機能する

記憶部 220 から必要な情報を取得し、庁書類提出タスクに設定するメッセージの内容や、当該タスクを割り当てる業務担当者を動的に変化させる。

[0475] 庁書類提出タスクが業務担当者に割り当てられると、当該業務担当者に通知される。通知を受信した業務担当者は、庁書類提出タスクに設定された内容に基づいて、上述した庁書類生成用のフローが生成した年金納付書の特許庁に提出する。業務担当者は、庁書類の提出を完了すると、庁書類提出タスクを完了する。庁書類提出タスクが完了すると、当該タスクの完了をトリガとして `Power Automate` に登録した庁書類提出プロセスノード完了処理用のフローが起動する。庁書類提出プロセスノード完了処理用のフローは、プロセスネットワーク状態管理部 260 の API にプロセスノード L27 のプロセス状態 D23 を完了状態に変更するプロセス状態変更要求を送信する。

[0476] 図 73 を参照して、請求プロセスノード L14 は、請求書生成プロセスノード L28 および請求書送付プロセスノード L29 を子として有する。

[0477] プロセスノード L28 は、請求書生成プロセスを表現する。請求書生成プロセスでは、年金納付業務に対する対価の請求を出願人に対して行うための請求書を生成する。請求書の生成は、`Power Automate` に登録した請求書生成用のフローを実行することによって行う。当該請求書生成用のフローの実行は、`Power Automate` の API に当該フローの実行要求を送信することによって行う。請求書生成用のフローは、`Power Automate for desktop` の請求内容入力用フローを起動する。`Power Automate for desktop` は、クライアント PC 上で動作する。`Power Automate for desktop` の請求内容入力用フローは、オンプレミスの会計システムに請求内容を入力する。

[0478] プロセスノード L28 のアクション情報 D253 は、`Power Automate` に登録した請求書生成用のフローを実行するための API アドレスを含む。また、プロセスノード L28 のアクション情報 D253 は、案件

番号等のコンテキストデータをAPIデータとして含む。プロセスネットワーク実行部270は、プロセスノードL28のアクション情報D253に含まれるAPIアドレスに対して、請求書生成用のフローの実行要求を送信するとともに、当該アクション情報D253に含まれるAPIデータも送信する。請求書生成用のフローは、送信されたAPIデータに含まれるコンテキストデータを使用して、データベースサーバーとして機能する記憶部220から必要な情報を取得し、請求内容を動的に変化させる。

[0479] プロセスノードL29は、請求書送付プロセスを表現する。請求書送付プロセスでは、請求書送付依頼を業務担当者に行う。請求書送付依頼は、Power Automateに登録した請求書送付依頼用のフローを実行することによって行う。当該請求書送付依頼用のフローの実行は、Power AutomateのAPIに当該フローの実行要求を送信することによって行う。請求書送付依頼用のフローは、MICROSOFT 365（登録商標）のタスク管理サービスであるPlannerに、請求書送付タスクを登録し、当該登録したタスクに業務担当者を割り当てる。業務担当者にタスクが割り当てられると、業務担当者に通知される。

[0480] プロセスノードL29のアクション情報D253は、Power Automateに登録した請求書送付依頼用のフローを実行するためのAPIアドレスを含む。また、プロセスノードL29のアクション情報D253は、案件番号等のコンテキストデータをAPIデータとして含む。プロセスネットワーク実行部270は、プロセスノードL29のアクション情報D253に含まれるAPIアドレスに対して、請求書送付依頼用のフローの実行要求を送信するとともに、当該アクション情報D253に含まれるAPIデータも送信する。請求書送付依頼用のフローは、送信されたAPIデータに含まれるコンテキストデータを使用して、データベースサーバーとして機能する記憶部220から必要な情報を取得し、請求書送付タスクに設定するメッセージの内容や、当該タスクを割り振る業務担当者を動的に変化させる。

[0481] 請求書送付タスクが業務担当者に割り当てられると、当該業務担当者に通

知される。通知を受信した業務担当者は、請求書送付タスクに設定された内容に基づいて、上述した請求書生成用のフローが生成した請求書を出願人に送付する。業務担当者は、請求書の送付を完了すると、請求書送付タスクを完了する。請求書送付タスクが完了すると、当該タスクの完了をトリガとしてPower Automateに登録した請求書送付プロセスノード完了処理用のフローが起動する。請求書送付プロセスノード完了処理用のフローは、プロセスネットワーク状態管理部260のAPIにプロセスノードL29のプロセス状態D23を完了状態に変更するプロセス状態変更要求を送信する。

[0482] 上述したプロセスマスタノード（プロセスノード）のうち、星印が付されているプロセスマスタノードML22、ML24、ML27およびML29（プロセスノードL22、L24、L27およびL29）の非同期フラグマスタ（非同期フラグ）の値はTrueであり、星印が付されていないプロセスマスタノード（プロセスノード）の非同期フラグマスタ（非同期フラグ）の値はFalseである。上述したプロセスマスタノード（プロセスノード）のうち、四角印が付されているプロセスマスタノードML24、ML27およびML29（プロセスノードL24、L27およびL29）の自動起動フラグマスタDM255（自動起動フラグD255）の値はFalseであり、星印が付されていないプロセスマスタノード（プロセスノード）の自動起動フラグマスタDM255（自動起動フラグD255）の値はTrueである。

[0483] 図74から図78を参照して、本実施例の動作を説明する。図74から図78は、実施例に係るプロセス実行システムにおいて実行される処理の一部を表すシーケンス図である。

[0484] 図74を参照して、プロセスネットワーク生成部250は、プロセスネットワークモデルDM2の入力を受け付けると、対応するプロセスネットワークD2を生成する。プロセスネットワーク生成部250のプロセスネットワークモデル入力部151は、コンテキストデータとして案件番号等を受け付け

ける。プロセスネットワーク生成部250は、プロセスネットワークモデル入力部151が受け付けたプロセスネットワークモデルDM2からプロセスネットワークD2を生成する際に、コンテキストデータを使用することによって、生成するプロセスネットワークD2のアクション情報D253に含まれるAPIデータに案件番号等のコンテキストデータを埋め込む。プロセスネットワーク生成部250は、生成したプロセスネットワークD2を記憶部220に記憶する。プロセスネットワーク生成部250は、生成したプロセスネットワークD2を記憶部220に記憶する際に、プロセスネットワークモデル入力部151が受け付けたコンテキストデータを当該プロセスネットワークD2と関連付けて記憶部220に保存する。

[0485] プロセスネットワーク状態管理部260は、プロセスノードの状態遷移処理を実行して、問い合わせメール生成プロセスノードL21を実行可能状態にする。生成された直後のプロセスネットワークD2に状態遷移処理を適用すると、問い合わせメール生成のプロセスノードL21が最初に実行可能状態になる。

[0486] 図75を参照して、プロセスネットワーク実行部270は、実行可能状態になった問い合わせメール生成プロセスノードL21を実行する。プロセスネットワーク実行部270は、Power AutomateのAPIに問い合わせメール生成用フローの実行要求を送信する。Power Automateの問い合わせメール生成用フローは、問い合わせメールを生成する。プロセスネットワーク実行部270は、問い合わせメール生成プロセスノードL21を実行した後に、問い合わせメール生成プロセスノードL21のプロセス状態D23を完了状態にするプロセス状態変更要求をプロセスネットワーク状態管理部260のAPIに送信する。プロセスネットワーク状態管理部260は、問い合わせメール生成プロセスノードL21のプロセス状態D23を完了状態にする。

[0487] プロセスネットワーク状態管理部260は、プロセスノードD20の状態遷移処理を実行して、問い合わせメール送信プロセスノードL22を実行可

能状態にする。

- [0488] プロセスネットワーク実行部270は、実行可能状態になった問い合わせメール送信プロセスノードL22を実行する。プロセスネットワーク実行部270は、Power AutomateのAPIに問い合わせメール送信依頼用フローの実行要求を送信する。Power Automateの問い合わせメール送信依頼用フローは、Plannerに、問い合わせメール送信タスクを登録し、当該登録したタスクに業務担当者を割り当てる。
- [0489] 問い合わせメール送信プロセスノードL22の非同期フラグの値はTrueである。そのため、プロセスネットワーク実行部270は、問い合わせメール送信プロセスノードL22の実行後に当該プロセスノードL22のプロセス状態D23を完了状態にしない（図27参照）。
- [0490] Plannerに登録した問い合わせメール送信タスクが完了すると、Power Automateの問い合わせメール送信プロセスノード完了処理用のフローが起動する。問い合わせメール送信プロセスノード完了処理用のフローは、問い合わせメール送信プロセスノードL22のプロセス状態D23を完了状態に変更するプロセス状態変更要求をプロセスネットワーク状態管理部260のAPIに送信する。プロセスネットワーク状態管理部260は、問い合わせメール送信プロセスノードL22のプロセス状態D23を完了状態にする。
- [0491] プロセスネットワーク状態管理部260は、プロセスノードの状態遷移処理を実行し、ドキュメント生成プロセス起動プロセスノードL23を実行可能状態にする。
- [0492] 図76を参照して、プロセスネットワーク実行部270は、実行可能状態になったドキュメント生成プロセス起動プロセスノードL23を実行する。プロセスネットワーク実行部270は、Power AutomateのAPIにドキュメント生成プロセス起動用のフローの実行要求を送信する。当該ドキュメント生成プロセス起動用のフローは、序書類生成プロセスノードL26のプロセス状態D23を実行可能状態にするプロセス状態変更要求を

プロセスネットワーク状態管理部260のAPIに送信する。プロセスネットワーク状態管理部260は、庁書類生成プロセスノードL26のプロセス状態D23を実行可能状態にする。また、ドキュメント生成プロセス起動用のフローは、請求書生成プロセスノードL28のプロセス状態D23を実行可能状態にするプロセス状態変更要求をプロセスネットワーク状態管理部260のAPIに送信する。プロセスネットワーク状態管理部260は、請求書生成プロセスノードL28のプロセス状態D23を実行可能状態にする。

[0493] プロセスネットワーク実行部270は、ドキュメント生成プロセス起動プロセスノードL23を実行した後に、ドキュメント生成プロセス起動プロセスノードL23のプロセス状態D23を完了状態にするプロセス状態変更要求をプロセスネットワーク状態管理部260のAPIに送信する。プロセスネットワーク状態管理部260は、ドキュメント生成プロセス起動プロセスノードL23のプロセス状態D23を完了状態にする。ドキュメント生成プロセス起動プロセスノードL23のプロセス状態D23が完了状態になると、問い合わせプロセスノードL11配下のプロセスノードL21、L22およびL23全てが完了状態になるので、プロセスネットワーク状態管理部260は、プロセスノードの状態遷移処理を実行して、問い合わせプロセスノードL11のプロセス状態D23を完了状態にする。

[0494] プロセスネットワーク実行部270は、実行可能状態になった庁書類生成プロセスノードL26を実行する。プロセスネットワーク実行部270は、Power AutomateのAPIに庁書類生成用フローの実行要求を送信する。Power Automateの庁書類生成用フローは、年金納付書を生成する。プロセスネットワーク実行部270は、庁書類生成プロセスノードL26を実行した後に、庁書類生成プロセスノードL26のプロセス状態D23を完了状態にするプロセス状態変更要求をプロセスネットワーク状態管理部260のAPIに送信する。プロセスネットワーク状態管理部260は、庁書類生成プロセスノードL26のプロセス状態D23を完了状態にする。

- [0495] プロセスネットワーク実行部270は、実行可能状態になった請求書生成プロセスノードL28を実行する。プロセスネットワーク実行部270は、Power AutomateのAPIに請求書生成用フローの実行要求を送信する。Power Automateの請求書生成用フローは、請求書を生成する。プロセスネットワーク実行部270は、請求書生成プロセスノードL28を実行した後に、請求書生成プロセスノードL28のプロセス状態D23を完了状態にするプロセス状態変更要求をプロセスネットワーク状態管理部260のAPIに送信する。プロセスネットワーク状態管理部260は、請求書生成プロセスノードL28のプロセス状態D23を完了状態にする。
- [0496] ここまでのプロセスノードの実行によって、問い合わせプロセスノードL11（および当該プロセスノードL11配下のプロセスノードL21、L22およびL23）、庁書類生成プロセスノードL26および請求書生成プロセスノードL28が完了状態になる。
- [0497] 問い合わせプロセスノードL11が完了状態になることによって、プロセスネットワーク状態管理部260による状態遷移処理によって次に実行可能状態になる可能性があるプロセスノードは指示確認プロセスノードL24である。しかしながら、指示確認プロセスノードL24の自動起動フラグはFalseであるため、問い合わせプロセスノードL11が完了状態になっても、プロセスネットワーク状態管理部260による状態遷移処理では指示確認プロセスノードL24は実行可能状態にならない。
- [0498] 同様に、庁書類提出プロセスノードL27の自動起動フラグもFalseであるため、庁書類生成プロセスノードL26が完了状態になっても、プロセスネットワーク状態管理部260による状態遷移処理では庁書類提出プロセスノードL27は実行可能状態にならない。さらに同様に、請求書送付プロセスノードL29の自動起動フラグもFalseであるため、請求書生成プロセスノードL28が完了状態になっても、プロセスネットワーク状態管理部260による状態遷移処理では請求書送付プロセスノードL29は実行

可能状態にならない。

- [0499] 図 7 7 を参照して、問い合わせメール送信プロセスノード L 2 2 によって表現されるプロセスにおいて送信された問い合わせメールに対する出願人からの指示メールを受信すると、Power Automate（登録）の指示確認プロセス起動処理用のフローが起動する。指示確認プロセス起動処理用のフローは、指示確認プロセスノード L 2 4 のプロセス状態 D 2 3 を実行可能状態にするプロセス状態変更要求をプロセスネットワーク状態管理部 2 6 0 の API に送信する。
- [0500] プロセスネットワーク実行部 2 7 0 は、実行可能状態になった指示確認プロセスノード L 2 4 を実行する。プロセスネットワーク実行部 2 7 0 は、Power Automate の API に指示確認依頼用フローの実行要求を送信する。Power Automate は、指示確認依頼用フローを実行して、MICROSOFT 3 6 5（登録商標）の承認依頼を業務担当者に対して送信する。指示確認プロセスノード L 2 4 の非同期フラグは True である。そのため、プロセスネットワーク実行部 2 7 0 は、指示確認プロセスノード L 2 4 の実行後に当該プロセスノード L 2 4 を完了状態にしない（図 2 7 参照）。
- [0501] 指示確認プロセスノード L 2 4 において送信した指示確認の承認依頼が承認されると、Power Automate（登録）の指示確認プロセスノード完了処理用のフローが起動する。指示確認プロセスノード完了処理用のフローは、指示確認プロセスノード L 2 4 のプロセス状態 D 2 3 を完了状態にするプロセス状態変更要求をプロセスネットワーク状態管理部 2 6 0 の API に送信する。プロセスネットワーク状態管理部 2 6 0 は、指示確認プロセスノード L 2 4 のプロセス状態 D 2 3 を完了状態にする。
- [0502] プロセスネットワーク状態管理部 2 6 0 は、プロセスノードの状態遷移処理を実行して、ドキュメント送信プロセス起動プロセスノード L 2 5 のプロセス状態 D 2 3 を実行可能状態にする。
- [0503] プロセスネットワーク実行部 2 7 0 は、実行可能状態になったドキュメン

ト送信プロセス起動プロセスノードL 2 5を実行する。プロセスネットワーク実行部2 7 0は、Power AutomateのAPIにドキュメント提出プロセス起動用のフローの実行要求を送信する。当該ドキュメント提出プロセス起動用のフローは、庁書類提出プロセスノードL 2 7のプロセス状態D 2 3を実行可能状態にするプロセス状態変更要求をプロセスネットワーク状態管理部2 6 0のAPIに送信する。プロセスネットワーク状態管理部2 6 0は、庁書類提出プロセスノードL 2 7のプロセス状態D 2 3を実行可能状態にする。また、ドキュメント提出プロセス起動用のフローは、請求書送付プロセスノードL 2 9のプロセス状態D 2 3を実行可能状態にするプロセス状態変更要求をプロセスネットワーク状態管理部2 6 0のAPIに送信する。プロセスネットワーク状態管理部2 6 0は、請求書送付プロセスノードL 2 9のプロセス状態D 2 3を実行可能状態にする。

[0504] プロセスネットワーク実行部2 7 0は、ドキュメント送信プロセス起動プロセスノードL 2 5を実行した後に、ドキュメント送信プロセス起動プロセスノードL 2 5のプロセス状態D 2 3を完了状態にするプロセス状態変更要求をプロセスネットワーク状態管理部2 6 0のAPIに送信する。プロセスネットワーク状態管理部2 6 0は、ドキュメント送信プロセス起動プロセスノードL 2 5のプロセス状態D 2 3を完了状態にする。ドキュメント送信プロセス起動プロセスノードL 2 5が完了状態になると、指示受領プロセスノードL 1 2配下のプロセスノードL 2 4およびL 2 5全てが完了状態になるので、プロセスネットワーク状態管理部2 6 0は、プロセスノードの状態遷移処理を実行して、指示受領プロセスノードL 1 2のプロセス状態D 2 3を完了状態にする。

[0505] 図7 8を参照して、プロセスネットワーク実行部2 7 0は、実行可能状態になった庁書類提出プロセスノードL 2 7を実行する。プロセスネットワーク実行部2 7 0は、Power AutomateのAPIに庁書類提出依頼フローの実行要求を送信する。Power Automateは、庁書類提出依頼フローを実行して、Plannerに庁書類提出タスクを登録

し、当該登録したタスクに業務担当者を割り当てる。庁書類提出プロセスノードL27の非同期フラグはTrueである。そのため、プロセスネットワーク実行部270は、庁書類提出プロセスノードL27の実行後に当該プロセスノードL27を完了状態にしない。

[0506] プロセスネットワーク実行部270は、実行可能状態になった請求書送付プロセスノードL29を実行する。プロセスネットワーク実行部270は、Power AutomateのAPIに請求書送付依頼用フローの実行要求を送信する。Power Automateは、請求書送付依頼用フローを実行して、Plannerに請求書送付タスクを登録し、当該登録したタスクに業務担当者を割り当てる。請求書送付プロセスノードL29の非同期フラグはTrueである。そのため、プロセスネットワーク実行部270は、請求書送付プロセスノードL29の実行後に当該プロセスノードL29を完了状態にしない。

[0507] Plannerに登録した庁書類提出タスクが完了すると、Power Automateの庁書類提出プロセスノード完了処理用のフローが起動する。庁書類提出プロセスノード完了処理用のフローは、庁書類提出プロセスノードL27のプロセス状態D23を完了状態にするプロセス状態変更要求をプロセスネットワーク状態管理部260のAPIに送信する。プロセスネットワーク状態管理部260は、庁書類提出プロセスノードL27のプロセス状態D23を完了状態にする。庁書類提出プロセスノードL27が完了状態になると、庁手続プロセスノードL13の配下のプロセスノードL26およびL27全てが完了状態になるので、プロセスネットワーク状態管理部260は、プロセスノードを状態遷移させる処理を実行して、庁手続プロセスノードL13のプロセス状態D23を完了状態にする。

[0508] Plannerに登録した請求書送付タスクが完了すると、Power Automateの請求書送付プロセスノード完了処理用のフローが起動する。請求書送付プロセスノード完了処理用のフローは、請求書送付プロセスノードL29のプロセス状態D23を完了状態にするプロセス状態変更要求

をプロセスネットワーク状態管理部260のAPIに送信する。プロセスネットワーク状態管理部260は、請求書送付プロセスノードL29のプロセス状態D23を完了状態にする。請求書送付プロセスノードL29が完了状態になると、請求書送付プロセスノードL14の配下のプロセスノードL28およびL29全てが完了状態になるので、プロセスネットワーク状態管理部260は、プロセスノードを状態遷移させる処理を実行して、請求プロセスノードL14のプロセス状態D23を完了状態にする。

[0509] 以上の処理で、ルートプロセスノードL01の配下のプロセスノードL11、L12、L13およびL14全てが完了状態になるので、プロセスネットワーク状態管理部260は、プロセスノードを状態遷移させる処理を実行して、ルートプロセスノードL01のプロセス状態D23を完了状態にする。

[0510] 上述したように、本実施例に係るプロセス実行システム200によれば、複雑なビジネスプロセス、特に非同期処理を含むような複雑なビジネスプロセスであっても実行可能である。特に、階層型のプロセスネットワークモデルは、階層を深くすることでプロセスを細分化していくことができるため表現力が高く、階層型のプロセスネットワークモデルを受け付け可能に構成されたプロセス実行システム200は、複雑なビジネスプロセスであっても実行可能である。業務内容の変更には、プロセスネットワークモデルを変更することで対応可能であり、プロセスネットワーク生成部の構成、プロセスネットワーク状態管理部の構成およびプロセスネットワーク実行部の構成などは変更する必要がない。そのため、プロセス実行システム200は、シンプルな構成でありながら可塑性と安定性に優れており、変化の速度が速いビジネス環境への適応度が高いシステムである。

[0511] 以上に説明したプロセス実行システムの構成は、上述の実施形態の特徴を説明するにあたって主要構成を説明したのであって、上述の構成に限られず、特許請求の範囲内において、種種改変することができる。また、一般的なプロセス実行システムが備える構成を排除するものではない。

[0512] 上述した実施形態では、プロセス実行システムは、サーバーマシン上で動作するWebアプリとして構成されたが、PC上で動作するネイティブアプリとして構成されてもよい。

[0513] 上述した実施形態はビジネスプロセスに対して本発明を適用した場合を説明したが、本発明はビジネスプロセスに限らず、例えば、CPS（Cyber-Physical Systems）におけるプロセスのように、現実世界との相互作用を含むプロセスにも適用可能である。

符号の説明

[0514] 100、200、300、400、500 プロセス実行システム
110、210、310、410、510 制御部
150、250、350、450 プロセスネットワーク生成部
151、351、451 プロセスネットワークモデル入力部
452 プロセスネットワークモデル出力部
160、260、360 プロセスネットワーク状態管理部
261 プロセス状態変更要求受付部
170、270、370 プロセスネットワーク実行部
380 転写部
381 転写要求受付部
382 転写因子モデル評価部
383 転写プロセス実行部
590 プロセスネットワークモデル表示部
591 プロセスネットワークモデル表示要求受付部
592 プロセスネットワークモデル表示データ生成部
595 プロセスネットワーク表示部
596 プロセスネットワーク表示要求受付部
597 プロセスネットワーク表示データ生成部
120、220 記憶部
130、230 通信部

140、240 バス

DM1、DM2、DM3、DM4、DM5 プロセスネットワークモデル

DM10、DM20、DM30、DM40、DM50 プロセスマスタノード

DM11、DM21、DM31 プロセスマスタノード識別子

DM12、DM22、DM32 親プロセスマスタノード識別子

DM15、DM25、DM35 プロセス生成情報

DM151、DM251、DM351、DM451 プロセス番号マスタ

DM152、DM252、DM352 プロセス名マスタ

DM153、DM253、DM353 アクション情報マスタ

DM1531、DM2531 アクションアイテムマスタ

DM254、DM354、DM554 非同期フラグマスタ

DM255、DM355、DM555 自動起動フラグマスタ

DM356 活性フラグマスタ

PDinfo 転写プロセス定義情報

PT 転写プロセスタグ

PTs 選択プロセスタグ

PTi 反復プロセスタグ

PTr 置換プロセスタグ

MDinfo 転写因子モデル定義情報

MT 転写因子モデルタグD1、D2、D3、D4、D5 プロセスネットワーク

D10、D20、D30、D40、D50 プロセスノード

D11、D21、D31 プロセスノード識別子

D12、D22、D32 親プロセスノード識別子

D13、D23、D33、D53 プロセス状態

D15、D25、D35 プロセス定義情報

D 1 5 1、D 2 5 1、D 3 5 1 プロセス番号
D 1 5 2、D 2 5 2、D 3 5 2 プロセス名
D 1 5 3、D 1 5 3、D 3 5 3 アクション情報
D 1 5 3 1、D 2 5 3 1 アクションアイテム
D 2 5 4、D 3 5 4 非同期フラグ
D 2 5 5、D 3 5 5 自動起動フラグ
D 3 5 6 活性フラグ

C I D ケース I D

C I D 1 アプリケーション名

C I D 2 テーブル名

C I D 3 レコード I D

C x D コンテキストデータ

C x T コンテキストテーブル

E 環境

E i n 内部環境

E o u t 外部環境

M 転写因子モデル

D i n 入力データ

D o u t 出力データ

R 評価結果

M e 評価対象転写因子モデル

M o 原評価対象転写因子モデル

シリアルライズしたデータ M D o u t

S y s O u t 外部システム

S 1、S 2 システム

T i n f o 転写情報

5 0 アクション実行システム

9 0 通信手段 C D ケースデータ

C D T ケースデータテーブル

請求の範囲

[請求項1]

プロセスの内容を定義したプロセス定義情報を生成するために使用されるプロセス生成情報を有するプロセスマスタノードを複数含み、当該複数のプロセスマスタノードによって生成されるプロセスノード同士の間を表現したプロセスネットワークモデルから、前記プロセス定義情報を有するプロセスノードを複数含むプロセスネットワークを生成するプロセスネットワーク生成部と、

前記プロセスネットワークに含まれるプロセスノードの状態と、所定のプロセス状態遷移規則と、に基づいて、特定のプロセスノードを実行可能にするプロセスネットワーク状態管理部と、

前記プロセスネットワークに含まれる実行可能な前記プロセスノードが有する前記プロセス定義情報によって定義されたプロセスを実行するプロセスネットワーク実行部と、を有し、

前記プロセスネットワーク生成部は、前記プロセスネットワークモデルの入力を受け付けるプロセスネットワークモデル入力部を備え、当該プロセスネットワークモデル入力部が入力を受け付けた前記プロセスネットワークモデルから前記プロセスネットワークを生成し、

前記プロセス定義情報は、当該プロセス定義情報によって定義されたプロセスを実行するのに必要なアクションに関する情報を含み、

前記プロセスネットワーク実行部は、要求に応じて所定のアクションを実行するアクション実行システムに対して、実行可能な前記プロセスノードの前記プロセス定義情報が含む前記アクションに関する情報に基づいて当該アクションの実行要求を送信することによって、当該プロセス定義情報によって定義されたプロセスを実行する、プロセス実行システム。

[請求項2]

前記プロセスネットワークモデル入力部は、前記複数のプロセスマスタノードによって生成される前記プロセスノード同士の間を階層的に表現した階層型のプロセスネットワークモデルの入力を受け

付け、

前記プロセスネットワーク生成部は、前記プロセスネットワークモデル入力部が入力を受け付けた前記階層型のプロセスネットワークモデルに対応する階層型のプロセスネットワークを生成する、請求項1に記載のプロセス実行システム。

[請求項3]

前記プロセスネットワーク状態管理部は、特定のプロセスノードの状態を変更するプロセス状態変更要求を受け付けるプロセス状態変更要求受付部を備え、当該プロセス状態変更要求受付部が受け付けた前記プロセス状態変更要求に基づいて前記プロセスネットワークに含まれる当該特定のプロセスノードの状態を変更する、請求項2に記載のプロセス実行システム。

[請求項4]

一の前記プロセスマスタノードが有する前記プロセス生成情報を、当該一の前記プロセスマスタノードに対応する一の前記プロセスノードに前記プロセス定義情報として転写する転写プロセスを実行する転写部をさらに有し、

前記プロセス生成情報は、前記転写プロセスを定義した一または複数の転写プロセス定義情報を含み、

前記転写プロセス定義情報は、プロセス実行システムが通信可能な環境が前記転写プロセスに与える影響をモデル化した転写因子モデルを定義した一または複数の転写因子モデル定義情報を含み、

前記転写部は、前記転写プロセスの実行要求を受け付ける転写要求受付部と、前記転写因子モデル定義情報によって定義された前記転写因子モデルを評価する転写因子モデル評価部と、前記転写プロセス定義情報によって定義された前記転写プロセスを実行する転写プロセス実行部と、を備え、

前記転写プロセス実行部は、前記転写因子モデル評価部が評価した一または複数の前記転写因子モデルの評価結果に基づいて前記転写プロセスを制御する、請求項3に記載のプロセス実行システム。

- [請求項5] 前記プロセスネットワーク実行部は、前記プロセス定義情報によって定義されたプロセスを実行する際に、前記転写プロセスの実行要求を前記転写要求受付部に送信することによって、当該プロセス定義情報を生成するために使用される前記プロセス生成情報を前記転写部が転写することによって更新された前記プロセス定義情報に基づいて当該プロセスを実行する、請求項4に記載のプロセス実行システム。
- [請求項6] 前記転写因子モデル評価部は、一の前記転写因子モデルを評価する際に、他の前記転写因子モデルの評価結果を当該一の前記転写因子モデルの入力データとして使用する、請求項5に記載のプロセス実行システム。
- [請求項7] 前記プロセス定義情報は、当該プロセス定義情報を有するプロセスが活性であるか非活性であるかを表現する活性フラグを含み、
前記プロセス生成情報は、前記活性フラグを生成するために使用される活性フラグマスタを含み、
前記活性フラグマスタは、前記転写因子モデルの評価結果に応じて前記活性フラグが定まるように構成された前記転写プロセス定義情報を含み、
前記プロセスネットワーク状態管理部は、前記プロセスノードが有する前記プロセス定義情報に含まれる前記活性フラグが非活性であることを表現している場合には、当該プロセスノードを存在しないものとして扱う、請求項6に記載のプロセス実行システム。
- [請求項8] 前記プロセスネットワーク生成部は、前記プロセスネットワークモデルを出力するプロセスネットワークモデル出力部を有し、
前記プロセスネットワークモデル出力部は、前記プロセスネットワークモデルをシリアルライズしたデータを出力する、請求項7に記載のプロセス実行システム。
- [請求項9] 前記プロセスネットワークモデル入力部は、前記プロセスネットワークモデルをシリアルライズしたデータの入力を受け付け、

前記プロセスネットワーク生成部は、前記プロセスネットワークモデル入力部が入力を受け付けた前記プロセスネットワークモデルをシリアルライズしたデータをデシリアルライズすることによって得られた前記プロセスネットワークモデルを用いて前記プロセスネットワークを生成する、請求項8に記載のプロセス実行システム。

[請求項10] 前記プロセスネットワークモデル出力部は、前記プロセスネットワークモデルをシリアルライズする際に、前記プロセスネットワークモデルと、当該プロセスネットワークモデルをシリアルライズしたデータとが一对一に対応する規則に基づいてシリアルライズする、請求項9に記載のプロセス実行システム。

[請求項11] 前記プロセスネットワークモデル出力部が前記プロセスネットワークモデルをシリアルライズしたデータを出力する際に使用するデータ形式は、キーと値の組を配列したデータ形式である、請求項10に記載のプロセス実行システム。

[請求項12] 前記プロセスネットワークモデル出力部が前記プロセスネットワークモデルをシリアルライズしたデータを出力する際に使用するデータ形式は、JSONである、請求項11に記載のプロセス実行システム。

[請求項13] 前記プロセスネットワークモデルを表示するプロセスネットワークモデル表示部をさらに有する、請求項4から12のいずれか1項に記載のプロセス実行システム。

[請求項14] 前記プロセスネットワークを表示するプロセスネットワーク表示部をさらに有する、請求項4から12のいずれか1項に記載のプロセス実行システム。

[請求項15] プロセス実行システムに用いられるデータ構造であって、
プロセスの内容を定義したプロセス定義情報を有するプロセスノードを複数含むプロセスネットワークを生成するために使用されるプロセスネットワークモデルを含み、
前記プロセスネットワークモデルは、前記プロセス定義情報を生成

するために使用されるプロセス生成情報を有するプロセスマスタノードを複数含み、当該複数のプロセスマスタノードによって生成されるプロセスノード同士の間の関係を表現し、

前記プロセス定義情報は、当該プロセス定義情報によって定義されたプロセスを実行するのに必要なアクションに関する情報を含む、データ構造。

[請求項16] 前記プロセスネットワークモデルは、前記複数のプロセスマスタノードによって生成される前記プロセスノード同士の間の関係を階層的に表現した階層型のプロセスネットワークモデルであり、当該階層型のプロセスネットワークモデルに対応する階層型のプロセスネットワークを生成するために使用される、請求項15に記載のデータ構造。

[請求項17] 前記プロセス生成情報は、当該プロセス生成情報を、当該プロセス生成情報を有する前記プロセスマスタノードに対応する前記プロセスノードに前記プロセス定義情報として転写する転写プロセスを定義した一または複数の転写プロセス定義情報を含み、

前記転写プロセス定義情報は、前記プロセス実行システムが通信可能な環境が前記転写プロセスに与える影響をモデル化した転写因子モデルを定義した一または複数の転写因子モデル定義情報を含む、請求項16に記載のデータ構造。

[請求項18] 一の前記転写因子モデルが他の前記転写因子モデルの評価結果を入力データとして設定可能に表現される、請求項17に記載のデータ構造。

[請求項19] 前記プロセス定義情報は、当該プロセス定義情報を有するプロセスが活性であるか非活性であるかを表現する活性フラグを含み、

前記プロセス生成情報は、前記活性フラグを生成するために使用される活性フラグマスタを含み、

前記活性フラグマスタは、前記転写因子モデルの評価結果に応じて前記活性フラグが定まるように構成された前記転写プロセス定義情報

を含む、請求項18に記載のデータ構造。

[請求項20] キーと値の組を配列したデータ形式によって表現される、請求項19に記載のデータ構造。

[請求項21] JSONのデータ形式によって表現される、請求項20に記載のデータ構造。

[請求項22] プロセスの内容を定義したプロセス定義情報を生成するために使用されるプロセス生成情報を有するプロセスマスタノードを複数含み、当該複数のプロセスマスタノードによって生成されるプロセスノード同士の間を関係を表したプロセスネットワークモデルの入力を受け付けるステップと、

前記プロセスネットワークモデルの入力を受け付けるステップにおいて入力を受け付けた前記プロセスネットワークモデルから、前記プロセス定義情報を有する前記プロセスノードを複数含むプロセスネットワークを生成するステップと、

前記プロセスネットワークに含まれる前記プロセスノードの状態と、所定のプロセス状態遷移規則と、に基づいて前記プロセスノードの状態遷移処理を行うステップと、

前記プロセスネットワークに含まれる実行可能な前記プロセスノードの実行処理を行うステップと、

を含む処理を、コンピュータに実行させるためのプロセス実行プログラム。

[請求項23] 前記プロセスネットワークモデルの入力を受け付けるステップにおいて、前記複数のプロセスマスタノードによって生成される前記プロセスノード同士の間を関係を階層的に表現した階層型のプロセスネットワークモデルの入力を受け付け、

前記プロセスネットワークを生成するステップにおいて、前記プロセスネットワークモデルの入力を受け付けるステップにおいて入力を受け付けた前記階層型のプロセスネットワークモデルに対応する階層

型のプロセスネットワークを生成する、請求項 2 2 に記載のプロセス実行プログラム。

[請求項24] 前記プロセスノードの状態遷移処理を行うステップは、特定のプロセスノードの状態を変更するプロセス状態変更要求を受け付けるステップと、当該プロセス状態変更要求を受け付けるステップにおいて受け付けた前記プロセス状態変更要求に基づいて前記プロセスネットワークに含まれる当該特定のプロセスノードの状態を変更する、請求項 2 3 に記載のプロセス実行プログラム。

[請求項25] 前記プロセス生成情報は、当該プロセス生成情報を、当該プロセス生成情報を有する前記プロセスマスタノードに対応する前記プロセスノードに前記プロセス定義情報として転写する転写プロセスを定義した一または複数の転写プロセス定義情報を含み、

前記転写プロセス定義情報は、プロセス実行システムが通信可能な環境が前記転写プロセスに与える影響をモデル化した転写因子モデルを定義した一または複数の転写因子モデル定義情報を含み、

前記転写プロセスの実行要求を受け付けるステップと、

前記転写因子モデル定義情報によって定義された前記転写因子モデルを評価するステップと、

前記転写プロセス定義情報によって定義された前記転写プロセスを実行するステップと、をさらに有し、

前記転写プロセスを実行するステップは、前記転写因子モデルを評価するステップにおける一または複数の前記転写因子モデルの評価結果に基づいて前記転写プロセスを制御する、請求項 2 4 に記載のプロセス実行プログラム。

[請求項26] 前記プロセスノードの実行処理を行うステップは、前記転写プロセスの実行要求を送信することによって、当該プロセス定義情報を生成するために使用される前記プロセス生成情報を転写することによって更新された前記プロセス定義情報に基づいて当該プロセスを実行する

、請求項 25 に記載のプロセス実行プログラム。

[請求項27] 前記転写因子モデルを評価するステップにおいて、一の前記転写因子モデルを評価する際に、他の前記転写因子モデルの評価結果を当該一の前記転写因子モデルの入力データとして使用する、請求項 26 に記載のプロセス実行プログラム。

[請求項28] 前記プロセス定義情報は、当該プロセス定義情報を有するプロセスが活性であるか非活性であるかを表現する活性フラグを含み、

前記プロセス生成情報は、前記活性フラグを生成するために使用される活性フラグマスタを含み、

前記活性フラグマスタは、前記転写因子モデルの評価結果に応じて前記活性フラグが定まるように構成された前記転写プロセス定義情報を含み、

前記プロセスノードの状態遷移処理を行うステップにおいて、前記プロセスノードが有する前記プロセス定義情報に含まれる前記活性フラグが非活性であることを表現している場合には、当該プロセスノードを存在しないものとして扱う、請求項 27 に記載のプロセス実行プログラム。

[請求項29] 前記プロセスネットワークモデルを出力するステップをさらに有し、

当該プロセスネットワークモデルを出力するステップにおいて、前記プロセスネットワークモデルをシリアル化されたデータを出力する、請求項 28 に記載のプロセス実行プログラム。

[請求項30] 前記プロセスネットワークモデルの入力を受け付けるステップにおいて、前記プロセスネットワークモデルをシリアル化されたデータの入力を受け付け、

前記プロセスネットワークを生成するステップにおいて、前記プロセスネットワークモデルの入力を受け付けるステップにおいて入力を受け付けた前記プロセスネットワークモデルをシリアル化されたデー

タをデシリアライズすることによって得られた前記プロセスネットワークモデルを用いて前記プロセスネットワークを生成する、請求項29に記載のプロセス実行プログラム。

〔請求項31〕 前記プロセスネットワークモデルを出力するステップにおいて、前記プロセスネットワークモデルをシリアライズする際に、前記プロセスネットワークモデルと、当該プロセスネットワークモデルをシリアライズしたデータとが一对一に対応する規則に基づいてシリアライズする、請求項30に記載のプロセス実行プログラム。

〔請求項32〕 前記プロセスネットワークモデルを出力するステップにおいて、前記プロセスネットワークモデルをシリアライズしたデータを、キーと値の組を配列したデータ形式によって出力する、請求項31に記載のプロセス実行プログラム。

〔請求項33〕 前記プロセスネットワークモデルを出力するステップにおいて、前記プロセスネットワークモデルをシリアライズしたデータを、JSONのデータ形式によって出力する、請求項32に記載のプロセス実行プログラム。

〔請求項34〕 前記プロセスネットワークモデルを表示するステップをさらに有する、請求項25から請求項33のいずれか1項に記載のプロセス実行プログラム。

〔請求項35〕 前記プロセスネットワークを表示するステップをさらに有する、請求項25から請求項33のいずれか1項に記載のプロセス実行プログラム。